

# 仮想電子計算機と計算機言語システム

—世界計量経済モデル分析システム—

安 田 聖 著

神 戸 大 学

経 済 経 営 研 究 所

1986

# 仮想電子計算機と計算機言語システム

—世界計量経済モデル分析システム—

安 田 聖 著

神戸大学経済経営研究所

1 9 8 6

## はじめに

1982年1月、電子計算機発祥の地であるペンシルバニア大学に出向き、プロジェクト・リンクの世界計量経済モデル分析システムの新規開発・維持を行う目的で約2年間滞在了。この滞在中、電子計算機自身のことを学ぶことはなかったが、システム概念については多くのことを学んだ。特に仮想電子計算機（IBMが商品化している仮想電子計算機とは概念を異にする）という概念は、今まで自分自身のシステム、特にパッケージ・プログラムに対して持っていた概念を根本的に覆すものであった。

これとは別に、当時プロジェクト・リンクは、結合する国の数を約70ヶ国に増やそうとしている時であった。システムから見れば、国が増えても基本的にはプログラムの構成上、問題がある訳ではなかったが、当時稼働していたシステムをそのまま使用するのであれば、電子計算機の能力（処理能力）に問題があるところまで来ていた。このため効率の良いシステムの必要性を痛感せざるにはいられなかった。もし効率ということを主眼におくのであれば、マルチ・タスクやマルチCPU等を考慮して開発されなければならない所まで来ていた。これは、逆に一般の電子計算機の知識だけでは、開発が不可能であることを意味していた。

これらの要求の中で、当時稼働していたシステムの維持とは別に、新たにシステムを開発する議論がなされたが、上述した理由（一般の電子計算機の知識だけでは、開発が不可能である）により現在まで開発されずに来ている。これら多くの議論のなかで、各国に一つのタスクもしくは一つのCPUを与え、貿易関連ブロックも定義式だけからなるモデルと見做して解く（京都大学東南アジア研究センターのシステムでは、この方法を一部採用した）ことが可能であることを提案したが実現されずに終わった。しかし、ここ二、三年のパーソナル・

コンピュータの進歩は目覚ましいものがあり、当時の提案がタスクではなく各一台のパーソナル・コンピュータで実現可能なところまできている。

本文は、仮想電子計算機という概念を導入することより、パッケージ・プログラムや最近盛んに言われるようになった第四世代言語を統一的に説明することを試みた。特に、具体的例として計量経済モデル分析システムを取上げて説明した。さらに、各国を一つのタスク（または、CPU）で構成する場合の方法についても仮想電子計算機という概念を導入することにより説明することを試みた。

本文の下書きに目を通して、仮想電子計算機という概念を更に発展させていけば現在、更に近い将来実現されるであろう電子計算機システムを総て説明することが可能であることを、自分自身が考えていたよりさらに大きく展開する貴重な助言を与えて下さった和歌山大学の杉浦一平教授に感謝の意を表したい。

1986年1月

安 田 聖

# 目 次

はじめに

序章 電子計算機とその利用言語	1
第一章 電子計算機言語の構造	7
1.1 応用専用計算機言語	7
1.2 ファームウェアの発達とシステム設計	10
1.3 計算機言語の設計	14
1.3.1 翻訳とインタプリタ	14
1.3.2 仮想電子計算機	15
1.3.3 専用言語電子計算機	19
1.4 計量経済分析のための仮想電子計算機	21
第二章 計量経済分析システム	23
2.1 計量経済分析システムの機能	23
2.1.1 計量経済モデルの構造の特化	24
2.1.2 データ整備と計量経済モデルの推定	26
2.1.3 モデル・シミュレーション	26
2.2 計量経済分析システムの構造	28
2.2.1 必要なデータの準備	28
2.2.2 モデルのプログラムへの組み込み	29
2.2.3 モデルの解法	31
第三章 計量経済分析システムの具体例	33
3.1 ペンシルバニア大学のシステム	34

3.1.1	BANKシステム	37
3.1.2	AUTOシステム	45
3.1.3	SIMDATシステム	46
3.1.4	SIMシステム	46
3.1.5	LINDATシステム	50
3.1.6	LINKシステム	50
3.2	計量経済分析言語	54
3.2.1	TSPとSTEPSシステム	54
3.2.2	STSとIASシステム	55
第四章 世界経済モデルの構造とその解法		61
4.1	世界経済モデル	61
4.1.1	各国別モデル	62
4.1.2	簡単なモデル	62
4.1.3	直接連関ブロックを含む複雑なモデル	66
4.2	世界経済モデルの解法	67
4.2.1	第一の方法	67
4.2.2	第二の方法	68
4.2.3	第三の方法	71
4.3	世界モデルの結合編集	72
第五章 世界経済モデルの解法システム		75
5.1	単一モデル解法システムと世界モデル解法システム	75
5.2	世界経済モデル・システムの作成	78
5.3	世界計量経済モデル分析システムの開発手順	80

第六章 東南アジア研究センターのシステム	89
6.1 各国別モデルの準備	89
6.2 解の順番の指定	90
6.3 データ転送用テーブル	93
第七章 分散型仮想電子計算機	95
7.1 複合体仮想電子計算機	95
7.2 データベース・マネージメント・システム	97
7.3 分散型仮想電子計算機	99
7.4 分散型計量経済モデル分析システム	100
7.4.1 国別モデル仮想電子計算機群	101
7.4.2 ワールド、地域モデル仮想電子計算機	102
7.5 分散型計量経済モデル分析システムの機能	102
第八章 おわりに	107
参 考 文 献	108
付 録	
TSPによるクラインの第一モデルの例	111
STSによるクラインの第一モデルの例	123

## 序章 電子計算機とその利用言語

1946年米国のペンシルバニア大学のムーア・スクールで最初の計数型電子計算機（現在のデジタル式電子計算機）ENIAC（Electronic Numerical Integrator And Computer）が稼働し始めてから40年の歳月が過ぎた。この短い期間にこれほど劇的な発展を遂げ、これほど大きい影響を社会、経済、科学、文化のあらゆる側面に与えたものは、他に殆ど例を見ない。かつて動力機を中心とする一連の機械の出現が工業経済社会という全く新しい社会を出現させた様に、電子計算機と、これに関連する通信を始めとする広範な分野での進歩が、社会の政治・経済機構の全面的な革命をもたらすに十分な力をもっていることは、今や、明確な事実となってきた。

その間、使用素子も真空管からトランジスタ、IC、LSIそしてVLSIと変化し、プログラムの作成方法も電気回路によるものからプログラム内蔵方式へと変って、大型化の一途をたどってきた。また利用形態もカード入力方式によるバッチ形式から、リモート・ターミナル方式によるTSS（Time Sharing System）、さらにインテリジェント・ターミナル方式による分散処理へと変化を遂げた。情報を記憶する方式も磁気記憶方式の多様な進歩に加えて、光学的方式によるものが実用化され、膨大な量のあらゆる形態の情報、プログラム、数値、文字、図形、映像、音声・音響等々を総合的に処理することが可能になった。また、これらに伴って入力方法も、音声入力、図形入力等々が可能となってきた。

これら、素子、プログラム、記憶装置のあらゆる分野で、開発が白熱化しており、近い将来に、どのような画期的な方式が出現して実用化するかという予測すら許されない状態にある。しかも、既に実用化された現存技術に限っても、



その全面的な展開によってなし遂げられるであろう人間文明の変革の範囲は、われわれの想像力の限界をはるかに超えている。

これらの電子計算機が急速に大型化する一方で、1971年米国のインテル社から世界初のマイクロプロセッサが発表されて以来、今までの電子計算機の利用形態とは違う個人が占有して使用する新しい利用方法を見出すに至った。パーソナル・コンピュータの出現である。これは1980年以降の短期間に驚異的な目覚ましい発達を遂げ、初期の4ビットCPUから最近では32ビットCPUのものまで、卓上型電子計算機が実用化して来ている。そして、データ通信の飛躍的な進歩によって、パーソナル・コンピュータと大型電子計算機のインテリジェント・ターミナルとの境界は、特殊用途のものを除けば、殆ど取り払われてしまったと言える。さらに、パーソナル・コンピュータと大型機との境界すら、次第にあやふやになり始めた。

この様な情勢に伴って電子計算機を各方面に利用する努力も各分野でたゆまず行なわれ、急激にその範囲を拡大している。マイクロプロセッサを組み込んだカメラのレンズやカードに始って、携行用のハンドヘルド・コンピュータ、ワードプロセッサ専用機の様な個人的に使用するシステムから、超大型電子計算機を必要とする、電力系統の制御や、銀行特に為替交換のオンライン処理等の巨大なシステムにいたるまで、あらゆる分野で色々なシステムが急速かつ広範に開発されてきている。

これらのシステムは、各々独自の命令体系（コマンド体系）を持ち、機能も応用分野も千差万別であって画一的に比較するのは困難である。しかし電子計算機言語構造の角度から見れば、各々のシステムの構成は、システムを駆動するに当たって、それを制御するためにあたえられる命令（コマンド）の入力方式が、翻訳（コンパイラーまたはアセンブラー）形式かインタプリター（即時実行）形式か、あるいはパラメータ方式ないしメニュー方式か、又はこれらの組み合わせが何層にも渡っているものか、という点が違うだけである。

それらの命令のもつ機能、つまり、それによって作動するプログラムの機能は、個々には殆どが比較的単純なものだと言える。プログラマーの立場からは、ユーザー側のニーズにもとづいて与えられるインプットとアウトプットとの間を、これも与えられた処理方法に従って、汎用計算機言語を用いて記述するだけのことである。この局面ではプログラマーはいわば翻訳者としての役割を果たすにすぎない。もっとも翻訳といっても、ミスや脱落は論外として、原文よりも優れているものもあれば、間違っていないだけと言う様な水準のものもある。しかし、「一応、間違いがなければよい」という基準のもとでなら、出来て当たり前、という性質のものと言えるであろう。

システムの良否は、それに含まれていなければならない個々の機能に関する処理ルーティン（プログラム）には問題がないという前提のもとで、システム全体として矛盾や重複、不用部分、繁雑さ、あいまいさがないか、ユーザーにとって使いやすく、ユーザーが利用過程で抱くかもしれない疑問や、犯すかもしれないミスに対して、懇切な配慮がなされているか、等々の諸点が主なチェック・ポイントとなって判定される。

われわれの経験に照して言えば、大規模システムを設計する場合、まず、これを駆動するための言語体系を考えて、言語体系の整合性を、不必要な重複を除去した、必要且つ十分な命令文（Statements）のセットとして、考察することから始めるのが、最も効率的である。システムの目的を満たす整合的な命令文セットの検討には電子計算機の専門家でないユーザー側も、自己の専門的知識を背景に参加することが容易だし、電子計算機専門家が対象分野の十分な知識に欠けていることによって独善的な設計に陥る危険も避けることができる。そうして、このような命令文セットが得られた後は、各命令文の内容および容量を検討する。これには、電子計算機専門家のもつ知識と、対照分野の専門家の知識との十分な協同が必要である。対象となるハードウェアの性能の限界内で、最も効果的な設計の基礎は、この段階で決定される。この段階の作業が終

ると、それに基づくシステムの基本的部分の設計は、多くの場合、かなりルーティン的な性格の作業になる。

本文では、現存するシステムとこれから開発されるであろうシステムについて、一般的に電子計算機言語として捉えた場合における、システム設計の統一的な考え方を示すことを試みる。もちろん、現存する全てのシステムを考察することは、不可能なので、筆者が今まで専門としてきた計量経済分析システム、特に世界計量経済モデルに例をとって述べることにするが、本文に述べる考え方は、「計量経済分析システム」とあるのを各々の分野での専門のシステム名に読み変えることによって、同一の考え方を一般的に適用することが可能であると考えている。

更に、この考え方を、近年さかんに議論されているデータ・ベース・システムやそのアプリケーション言語（DBMS）にもとづくシステムに適用した場合について述べる。最後に、大型システムを分散処理システムとして構成する場合の一般的な考え方へと拡張することを試みた。

いまから四半世紀以上も前のことになるが、電子計算機の発展の初期、第一世代と呼ばれた頃には、平方根や対数等の初等関数を効率的に計算するプログラムの開発は、電子計算機科学の主要研究対象であった。今日ではシステム設計にあたって、これらを特にシステムの必要要素として意識する人はあるまい。行列演算についても同じことが言える。

今日のシステム設計で、システムの基本的な要素として考慮されるのは、例えば推定法について言えば、最小二乗法やコ克蘭・オーカット法と言った、より高次の範ちゅうの手法である。システム設計について議論するとき、これらのうち、どれとどれとをシステムに組み入れるかについて、また、それらを使うための命令の与え方や、結果をどの様にどの範囲で出力するかについて討議されるが、手法そのもののプログラム内容については、既知として、検討の対象にならないのが普通である。

むしろ、電子計算機の急速で広範な普及を反映して、システム設計においては、手法は既知として、電子計算機に不慣れなエンド・ユーザーにとって使いやすく、より包括的なシステムを提供することに、力点が置かれる様になってきている。

加えて、研究の大規模化とともに、多数の研究者による協同研究が普通に行われるようになり、世界的規模の協同研究すら珍しくはない状態になってきている。このような事情に対応するシステムの開発も、計画に上ってきている。本書では、このような情勢に対応しうる、システム設計の一般論を、考察の対象としたい。



# 第一章 電子計算機言語の構造

初期の電子計算機はそれを稼働するために、所望の計算手順にしたがって、スイッチやケーブルで計算回路を物理的に作って行かなければならなかった。プログラム内臓型（フォン・ノイマン型）の電子計算機の出現によって、オペレーション・コード（ある機能を実行させるための、一連のファンクション・スイッチの設定方法）を、2進数で表現し、連続する手順を8進数か16進数かで記述して、メモリーに入力し記憶させておいて連続実行させる、いわゆる機械語が誕生して、電子計算機言語の歴史が始った。

計量経済分析システムを電子計算機言語として捉える前に、まず、電子計算機とその言語の歴史的背景とその構造について考えてみる。

## 1.1 応用専用計算機言語

今から約40年前、1946年米国で最初の計数型電子計算機（現在のデジタル式電子計算機）ENIACが開発された。しかしこの電子計算機は、現在一般に使われているプログラム内蔵方式ではなく、ファンクション・スイッチによるプログラミング方式が採用されており、記憶装置も持たなかった。

かれらはその後、現在の電子計算機の原形とも言えるEDVAC（Electronic Discrete Variable Calculator）の開発に着手した。しかし、彼らとは別に1949年英国のケンブリッジ大学で、最初のプログラム内臓型の電子計算機となった、EDSAC（Electronic Delay Storage Automatic Computer）が先に開発された。

以後、電子計算機はその構成素子によって、真空管時代の第一世代、トランジスタ時代の第二世代、IC時代の第三世代、そしてLSI VLSI時代の第四

世代へと成長を遂げて来ている。またこれに対応して、各世代ごとに、電子計算機のプログラミングのための言語も、次第により自然言語に近いものが開発され、様々な水準と適用対象を想定した、アSEMBラー、インタプリター、コンパイラー、リンカー等が開発されてきた。その一方では、電子計算機の運用の効率化と利用の多様化に対応して、オペレーティング・システムが発達し、そのための言語が発展してきている。

電子計算機の機能に則して形成されている、これらの「汎用」言語が発達した一方で、電子計算機のあらゆる応用分野ごとに、その分野で成立している概念に則して形成された、「専用」言語が、電子計算機の処理能力と容量の大きさに対応して開発されてきている。

あらゆる応用分野における専用言語が、例えば、統計分析のためのSPSS (Statistical Package for Social Sciences) や SAS (Statistical Analysis System)。電気回路解析システムのECAP (Electronic Circuit Analysis Program)。データベース・システムの操作言語としての多数のエンド・ユーザー言語等々、広範囲に、かつ、多様に開発されている。

ディスプレイ端末が発達し、さらにVLSIの急激な進歩によって、端末のインテリジェンシィが高度化した。そして卓上型の強力な電子計算機であるパーソナル・コンピュータ、いわゆる「パソコン」が出現した。これらの発展は電子計算機のプログラミングの様相を一変させた。「会話型プログラミング」の誕生である。

汎用言語の分野では、パソコンに搭載されたマイクロソフト BASIC インタプリターの画期的な性能が数多くの専門家を熱狂させた。書いたプログラムを、すぐに、しかも、部分的にも、完成した分だけでも、簡単に実行できて確かめられる、会話型プログラミングは、非効率きわまりなかった旧来の一括入力方式のプログラミングとくらべて、まさに革命であった。インタプリターでプログラミングを行い、完成したものをコンパイラーに掛けて実行可能な機

械語レベルのモジュールをつくる方式は、C-言語をはじめ他の汎用言語におよびはじめている。

応用専用言語の分野では、会話型プログラミングの効果はより決定的であった。システムの側から出される指示にしたがって、枝分れ式に自分の行いたい処理を選択してゆけばよい、いわゆる「メニュー方式」は、電子計算機を専門家によって守護された神殿から開放して、一般の直接利用にゆだね、利用者人口の爆発的な増加を可能にした。

日本語ワードプロセッサ（ワープロ）も、アッという間に普及した。ワードプロセッシングという応用分野での専用言語であるこのシステムは、「手書き」という非効率性が宿命と思われて来た中国語、日本語、韓国語等の表意文字系言語に、タイプライティング可能な、その本来備えていたきわめて効率的な思想・意志の伝達手段としての価値を劇的に回復させた。表計算プログラムと、会話型のDBMSは、日本語ワープロと組合わされて、OA革命の一つの支柱になった。

ポータブル日本語ワープロ専用機も、KEY化されたメニュー言語をもつ固定用途の電子計算機である。この軽量で信じられぬくらい安価な電子計算機は、出現するや熱狂的な歓迎をうけ、平行して起った、ゲーム用のファミリー・コンピュータのブームとともに、いわゆるOA心身症が、日本の古い世代にかぎられた症候であり、情報化時代への日本人の適応能力の高さを確信させるに十分であった。

電子計算機の専門知識がなくても、各分野の専門家が自己の専門分野で電子計算機が自由に使用できる時代がすでに到来している。これを目的として開発される専用言語システムの数および質は、加速度的に向上するであろう。新しい専用言語システムを開発するための道具としての新しい言語も開発される様になってきた。計量経済分析システムもこれらの例外ではない。



## 1.2 ファームウェアの発達とシステム設計

電子計算機言語はそれがどんな目的のものであっても、それが稼働する電子計算機で実行可能な形の機械語に、何等かの形で変換されなければならない。汎用言語も専門分野別の専用言語も、実行可能な段階に到達するまでの処理過程からみた電子計算機言語構造上は、言語が処理される方法と性格に応じて、並列に組み分けされることになる。

初期の段階の電子計算機は、ハードウェア的には整数演算機構を持つにすぎず、乗除演算機構すらないものもあった。浮動小数点演算は電子計算機にとって不可欠ともいえる重要な機能だが、第一世代機では殆どがソフトによって処理していた。このようなより基本的・本質的な機能をハードウェア化することから始めて、IC、LSI、VLSIの進歩に伴い、今まで言語はソフトウェアとして電子計算機本体（ハードウェア）から分離して考えて来たのが、その境界が次第に明確で無くなりつつある。この境界領域に対してファームウェアと呼ばれる技術を適用して、電子計算機の処理能力の最適化を図るようになってきたのである。ファームウェアとは、一種のプログラムを電子計算機の中の書き換え不能な領域に電気回路として固定することによって、一連の機能を、見掛け上、電子計算機の新たな一命令として扱う技術である。

このファームウェア技術の適用が一般化し定着するに従って、電子計算機の設計方法にも変化をもたらした。つまり、電子計算機本体とそれを駆動する言語との間には、明確な区別がなくなり、新たに電子計算機を設計する場合、それによって主としてソフトウェアによって処理されると想定された言語をもハードウェアの側で考慮するようになってきたのである。パソコンにBASICがROMで搭載されたことが一般的であるように、O.S.や各種の言語プロセッサがチップ化されて供給されるようになるのも、あまり遠い将来ではない。

配列の演算を高速化するために、複数のプロセッサで平行処理する電子計算機も多数出現したし、各種のデバイスに専用のプロセッサとプログラム

ROMとを組みこんで、中央プロセッサのプログラムの負担を軽くする方式は、一般的になっている。

この傾向は当然に、電子計算機の用途別分類を、事務用、科学計算用といった大まかな分類から、より特殊な専門領域にまで拡大し細分化せずには済まない。VLSIの急速な進歩によって、電子計算機の製造コストが急激に低下してゆくことを背景に、専用言語の発達にともなって、その言語特性に示される特殊な処理方式に適したハードウェア特性をもつ専用電子計算機の開発が、近年活発に行われる様になってきた。汎用言語と専用言語とに分化した電子計算機言語の発展が、それに対応して、汎用電子計算機に対する専用電子計算機の発展を促したのである。

汎用機としてのパーソナル・コンピュータが、専用機としてのワード・プロセッサの爆発的な普及を促したように、または、LISPマシン、アレイ演算マシン、データ・ベース・マシン等々の平行処理計算機の開発が活発になって来たのに示される様に、将来、特定目的のために設計された専用電子計算機は、急速にその範囲を拡大してゆくであろう。

例えば、マイクロプロセッサを利用したパーソナル・コンピュータが汎用電子計算機として存在するのに対して、ある一連の機能を備えたプログラム・システムを与えればワード・プロセッサとして機能する。これをさらに進めて、ワード・プロセッシングを便利に行うための専用諸機能が、特に使い易くなる様に、ハードウェア的に特別に設計された専用電子計算機がワード・プロセッサ専用機である。その特徴は、汎用機用に組まれるワード・プロセッシング用のプログラムの相当部分をファームウェア化し、さらに、主要機能を刻印したKEYを特別に装置して、たいていの必要な機能は、その名称を打ったKEYを押すだけで実行されるようにした点にあるということができる。

システム設計者の立場からは、実現したい機能が、ソフトウェア的に実現されようが、ハードウェア的に実現されようが、どちらでもよいことである。

それをどのようなメディアで実現するかはハードウェア・エンジニアの領域であって、ソフトウェア・エンジニアあるいはシステム・エンジニアの責任領域ではない。彼らは既存の手段についてのリストのなかから、彼らの目的に一番適当と思われるものを選択すればよい。

今日、あるプログラム・システムを設計するにあたって、浮動小数点演算をどう取り扱うかを考える設計者は、まずいない。もっと複雑な概念についての処理方法を既知として、蓄積されて来た利用可能なソフトウェア群およびファームウェア群を前提にして出発するであろう。

電子計算機の設計方法の変化に示されるこの様な技術進歩に対応して、専用言語を、ハードウェアとしての電子計算機とソフトウェアとしての専用言語を「対」として一体化した「仮想電子計算機」(群)からなるシステム、として捉えるようになってきた。

電子計算機に読み込みさえすれば直ちに活性化して利用可能となる知識がプログラムの形で蓄積され累積してゆく。新しいシステムはこれら既知であり、既存である知識を結合することによって造り出される。新たに付加しなければならない部分は比較的少なく、それよりも新しい結合方法によって、利用者にとってより効率的で使い易いものを産み出すことに、より多くの努力が払われるのが、電子計算機のソフトウェア・システム開発の最近の特色である。設計にあたって思考の材料となる「利用可能な部品」である個々のソフトウェアまたはファームウェアを、そのシステムを構成するための「モジュール」と呼ぶことにすれば、システム設計者の要求する各モジュールは、時間の経過と共に急速にその種類、規模、機能および選択可能範囲を累積的に増大してゆく事を、最大の特徴としているのである。

例えば、逆行列の演算プログラムは20年前は数冊の専門書があったくらいの

問題であったが、今ならフローチャートに [INV] とでも書かれるだけだろう。それすら怪しいもので、もっと高度にインテグレートされた概念、例えば最小二乗法が、システムの構成要素の一単位として扱われ、[LSM] と書かれるだけに終わる、と言うのが実情であろう。あとは自分の調達できる各種の最小二乗法のプログラムのなかから今の目的に適當なものを選べばよい。

つまり、システム設計は電子計算機の固有の機能から出発するのではなく、プログラムされ、インテグレートされた機能から出発する。そしてその機能は時にソフトウェアではなくファームウェア化されていても、彼にとってはどちらでもよい事なのである。

これらのモジュールは、より基本的なものから次第にファームウェア化されて行く。このように考えてくると、システム・エンジニアにとって彼の設計の基礎となるモジュールは、将来ますます多くのハードウェア部分を含むようになるだろう。

さらに、通信回線が高速になり安価になって行けば、単一の電子計算機を前提にシステムを組むより、ネット・ワークで結合された一群の電子計算機を前提にシステムを考える方が、より効率的になってくる。システムの機能が多様な特性を含んでいる場合、それらを単一の電子計算機で実現するよりも、特殊機能をもった電子計算機を組み合わせるほうが、効率的であり、簡単である場合がおこってくる。

現在でも例えばプロッターは独自のCPUを持つ独自の電子計算機であり、あるシステムがプロッターによる描画を機能の中に含んでいるとき、システム・エンジニアはプロッターという専用電子計算機とそれに関して蓄積されているプロッター独自の言語で書かれたソフトウェア群とを、彼のモジュール・リストのなかに持って作業をしているのである。この場合、メインの電子計算機とサブの専用電子計算機であるプロッターとが、どのようにして給合されるかは、通常は彼の関心外の事である。何種類かのインターフェースがあり、そのどれ

かを使うのただだけ考えればよく、詳しいことは別の専門家に委せばよい。サブの電子計算機がメインの電子計算機とは遠隔の地点に設置されているとしても別に特別な考慮を要するわけではない。

この様を考えてくると、システム設計を行うにあたって、限られた特定の機能をもった専用電子計算機のセットを想定し、それを組み合わせて全体の構成を考えていくことが出来る。今後の電子計算機と通信ネット・ワークの発達を考えて、それを前提とした大規模システムの設計を行うには、単一の電子計算機を想定したソフトウェア・システムにこだわるのは、不利である。むしろ、ある機能を満たすモジュールを、ハードウェア、ソフトウェアを一体として考え、その組み合わせにおいてシステムを考えるほうが、今後の進歩によりスムーズに対応し得る。

### 1.3 計算機言語の設計

#### 1.3.1 翻訳とインタプリタ

汎用、専用を問わず電子計算機を新たに設計する場合、最初どんな命令（言語）体系を使用し、また何処までを電気回路で構成し、どこから先をソフトウェアでカバーするかを決めなければならない。一般に、コストや電子計算機の回路が複雑になるのを避けるため出来るだけ簡単な命令体系を採用する事が多い。通常これを機械語と呼ぶ。一方利用者側からすれば、より自然言語に近い命令体系を採用する事が望まれる。この場合、コストや電子計算機の回路が複雑になる。例えば、現在の技術水準からすれば、計量経済分析に必要な最小二乗法等を、LSMと言う機械語等として採用することも可能であるが、一般には、これら自然言語に近い命令体系を採用することは極く稀である。これら相反する要求を満足する為に現在まで多くの方法が考えられてきている。

いま実際には限られた範囲の機械語しか実行することができない電子計算機が、存在すると仮定しよう。以後説明のために、この電子計算機が実行する事が出来る機械語をL1とし、より自然語に近い命令体系をL2と呼ぶことにする。

第一の方法は、L2で書かれたプログラムを一命令ずつ同じ順序でL1の命令に置き換えていき、その後、完全にL1の命令に置き換えられた新しいプログラムを実行する方法である。

第二の方法は、L2で書かれたプログラムをL1で書かれたプログラムのデータとして読み込み、直接L2の命令を同一順序のL1の命令として実行する方法である。この方法は、第一の方法と違って、L1の命令を作成することはしない。

どちらの方法もL2の命令をL1の同一順序の命令として実行するという意味に於いては、よく似ている。前者はL2のプログラムを完全にL1のプログラムとして置き換え、その後、新しいL1のプログラムを実行する方法であるが、後者はL2のプログラムを直接実行するところが、違うだけである。前者の方法を翻訳（この中には、コンパイルとアセンブルの二種類の方法がある）と呼び、後者の方法をインタプリターと呼ぶ。

しかしこのL2の命令体系をしても、利用者にとっては自然言語とは言い難い場合が多い。このため、更にL2の上にL3という命令体系を、そしてそれでも不十分な場合、その上にL4……という命令体系をつくることになる。

この様に電子計算機を多層電子計算機として捉えるという提案は、最初1951年英国のM. V. Wilkes によってなされたが、その当時まだアセンブラーやコンパイラーが開発されなかった所で仮想電子計算機として捉えずに、ただ3層の電子計算機として捉えるにすぎなかった。

以下本文では、翻訳とインタプリター、そしてL1, L2, L3, L4……という命令（言語）体系についてより理解を深めるために、多層電子計算機を仮想電子計算機として捉えることにする。

### 1.3.2 仮想電子計算機

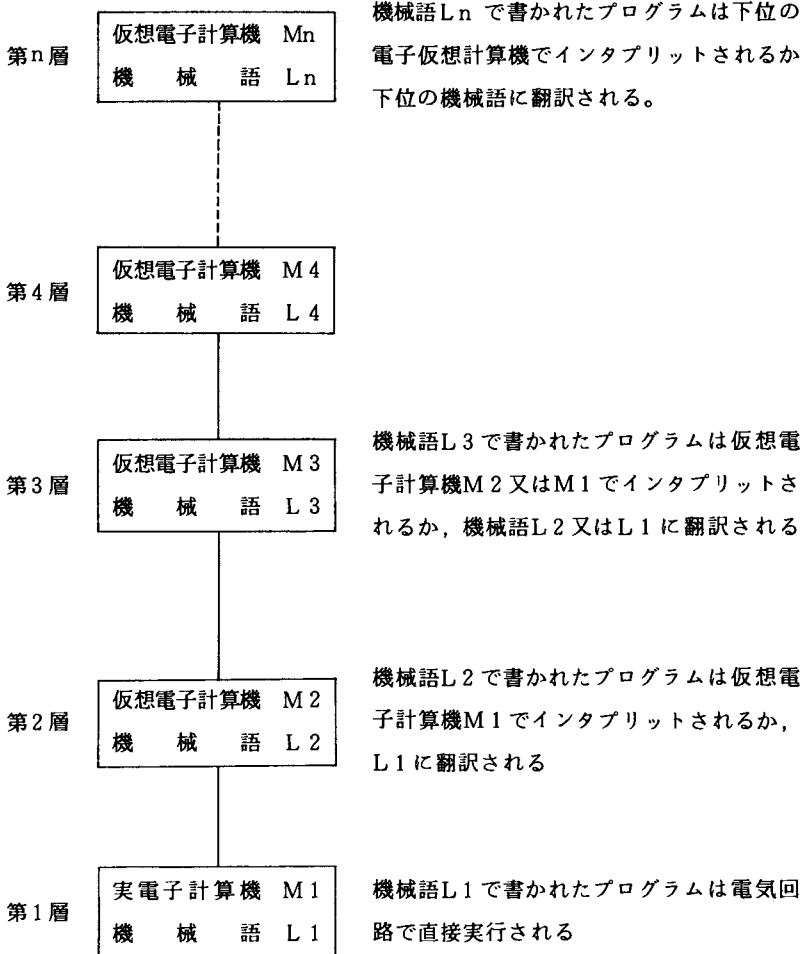
L1という命令（言語）体系を実行する電子計算機をM1とすると、L2という命令（言語）体系は前節で示した様に翻訳かインタプリターという手法を用いて、実際にはM1という電子計算機で実行されることになる。しかしこれは、

L2という命令体系が実行されるという意味において、L2という命令（言語）体系を機械語として実行する電子計算機が実在すると考えることができる。この実際には実在しない電子計算機を仮想電子計算機（Andrew S. Tanenbaum [1984]）と定義することにする。この仮想電子計算機をM2とすると、L3という命令（言語）体系は、M2という仮想電子計算機によって翻訳かインタプリターという手法を用いて実行される。同様にL4という命令（言語）体系はM3という仮想電子計算機によって、Lnという命令（言語）体系は、Mn-1という仮想電子計算機によって実行されると考えることができる。この様子を第1-1図に示す。

この様に、電子計算機言語を仮想電子計算機の言語として捉えると、各々の言語の種類によってこの仮想電子計算機の層数が異なり、また各層間の受渡しの方法が翻訳という手法を用いるのか、インタプリターという手法を用いるのかの違いによって分類する事が出来る。

第1-2図に最近の電子計算機の構成例を示す。近年設計される電子計算機の多くは、マイクロプログラミングと呼ばれる（機械語の更に下に位置するプログラムなのでこの様に呼ばれる）技術を使用して、一般に機械語と呼ばれる命令をマイクロプログラムと呼ばれるプログラムでインタプリットしている。このマイクロプログラミングを実行する電子計算機をデジタル回路レベルと呼ぶことにする。これが**実在する実電子計算機**である。この上にマイクロプログラミングレベルがあり、この上に一般に言われる機械語レベル、そしてオペレーティング・システム、アSEMBラー、問題向け言語レベルの電子計算機が実在することになる。つまり最下層のデジタル回路による電子計算機以外は、仮想的に作り上げた電子計算機であるため、これらを仮想電子計算機と呼ぶ。つまり、近年の機械語レベルの電子計算機も、その機能の全部が電気回路で構成されているのではなく、マイクロプログラミングを実行する電子計算機でインタプリットされている仮想電子計算機とみなすことが出来るのである。

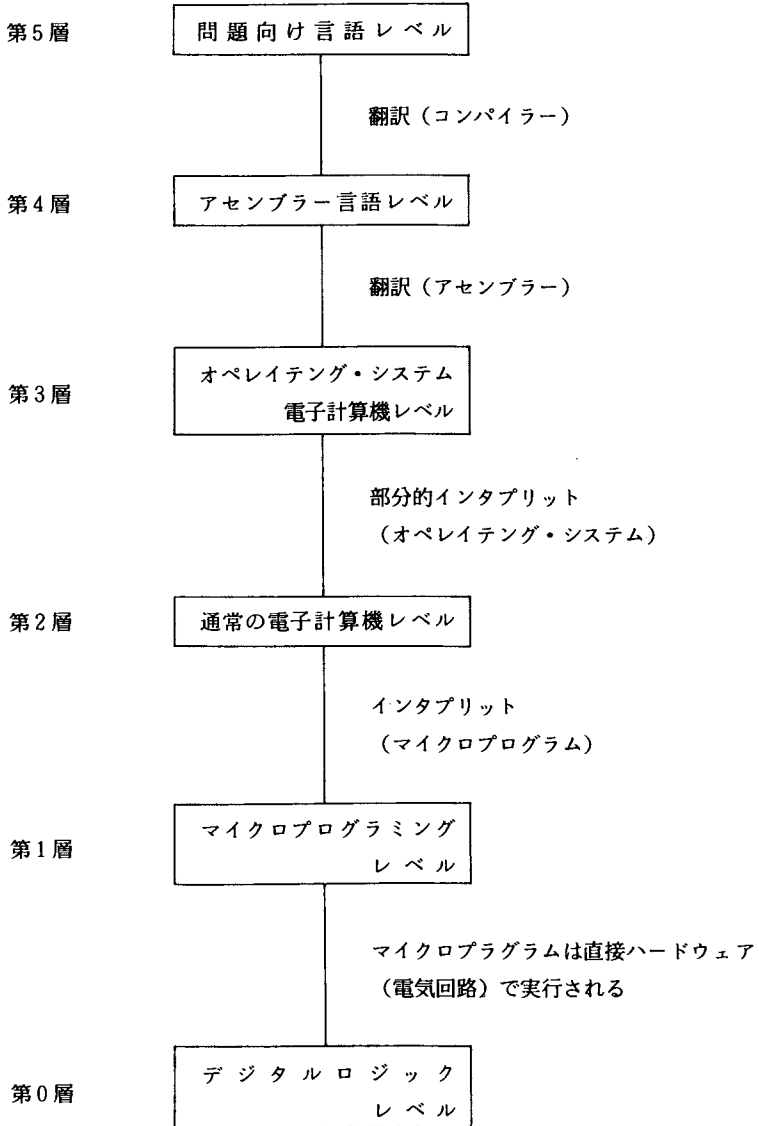
第 1 - 1 図 多層電子計算機



この考え方を、我々にとって最も馴染み深いFORTRAN言語について考えることにする。つまり、FORTRAN言語電子計算機の場合、この電子計算機が仮想電子計算機として、どの様に実現されているかを考えてみることにする。



第1-2図 最近の最も近代的な6層電子計算機



第1-2図によれば、FORTRAN言語レベルの仮想電子計算機は、第5層目に当たることになる。FORTRAN言語で書かれたプログラムは、FORTRAN翻訳プログラムによってアセンブラー言語に翻訳（コンパイラー）され、そしてオペレーティング・システム言語に翻訳（アセンブラー）される。前半の翻訳（コンパイル）が第4層レベルの仮想電子計算機に相当し、そして後半の翻訳（アSEMBル）が第3層レベルの仮想電子計算機に各々対応することになる。このオペレーティング・システム・レベルの言語は、さらに第1層の機械語電子計算機でもって直接実行され、一部はインタプリットされる。この機械語が、第1層（マイクロプログラミングレベル）の仮想電子計算機でインタプリットされる。さらに、このマイクロプログラムが、第0層の電子計算機で直接実行される。つまり、FORTRAN言語で記述されたプログラムは、一部は翻訳、一部はインタプリターと言う手法で何層にも渡って実行されることになる。

FORTRAN専用電子計算機（仮想電子計算機）は、FORTRAN言語を何ステップかに渡って機械語に翻訳し、その後マイクロプログラミングによってインタプリットする事によって実行される。

### 1.3.3 専用言語電子計算機

専用電子計算機の多くは、前節で説明した仮想電子計算機という考え方を導入する事によって簡単に構造を説明することが出来る。例えばFORTRAN言語によって記述された専用言語の場合、その専用言語がFORTRAN言語で記述されたシステムでインタプリットされるのか、FORTRAN言語で記述されたシステムによって他の言語（機械語の場合もあればBASIC, FORTRAN, アセンブラーの場合もありうる）に翻訳されるのかによって構成が異なることになる。あるシステムで他の言語から、FORTRAN言語等を生成する場合もある。一般に、あるシステムで他の言語からその言語とは異なった言語を生成する方法を、プレコンパイラーと呼ぶ。又場合によっては、FORTRAN言語で記述されたシステムが新たな言語に翻訳し、その言語を

FORTRANで記述された他のシステムがインタプリットする場合もある。つまり翻訳と実行が異なったシステムで行われる場合もある。

つまり専用電子計算機は、FORTRAN言語専用電子計算機の場合より、層数の多い仮想電子計算機で構成されていることになる。その層数は、その構成方法に依存していることになる。

この様に専用言語に対する専用電子計算機の構成方法は、それ自身単一のシステムで構成されている場合もあれば、複数のシステムで構成されている場合もあり、またその各々のシステムが何層にも渡って複数のシステムで構成される場合もあり、千差万別である。

- ・インタプリター方式の場合、即時実行という特徴を持つ代わりに  
実行するのに多くの情報を必要とする処理は苦手とする
- ・翻訳形式の場合、実行までに多くの時間を必要とする代わりに実行  
するのに多くの情報を必要とする処理（一括処理）を  
得意とする

等、多くの特徴、制約を持っている。それ故、構成方法によって自ずとそれ自身の機能が制約されることになる。換言すれば、目的に合った構成方法が存在することになる。

さらに、これら各層における処理を担当する各仮想電子計算機が、物理的に違った、互いに独立な電子計算機であってもよい。むしろ、分散処理の発達にともなって、各層別の各仮想電子計算機が別々のものであることが、むしろ当然になる傾向が強まるであろう。今日でも、一つのシステムをプログラムしてゆくのに、各部分システムの開発を、複数のインテリジェント端末（パソコン）を使用してFORTRAN, BASIC等、複数の言語を用いて独立に行い、その結合・編集と実行は大型機でおこなうのは、普通になっている。以上においては、電子計算機言語から出発して実行に至るまでを、縦に層化して、その各層について仮想電子計算機を考えたが、実行段階において、データ処理の形態に応じて、

並立的に選択できる各種手法と、処理の段階に対応して縦に層化された手法を考へて、一般的には格子状に配列された仮想電子計算機群として、一つのシステムを構想することが可能である。以下においては、このタイプのもものが考察の主対象となる。

#### 1.4 計量経済分析のための仮想電子計算機

電子計算機の進歩とともに各国で多くの計量経済分析システムが作成される様になってきた。代表的な計量経済分析システムとして国外では、米国のペンシルバニア大学とWEFAのBANK, AUTO, SIMDAT, SIMの一連のシステム, TSP (Time Series Processor), IBMのEPLANそしてTROLL, 西独のIAS (Inter-active Simulation System), オーストリアのSTS (Stochastic Simulation System), 国内では和歌山大学の杉浦一平, 神戸大学の定道宏らによるSTEPS (Statistical Techniques for Econometric Planning and Simulation), 慶応大学の森敬と通産省によるEMS (Econometric Modeling System), またパーソナルコンピュータ (マイクロプロセッサを使用した電子計算機の通称名) を使用したマイクロTSP, PC STEPSなど多くの例があげられる。これらは、各々独自の言語体系を持ち、各々機能も、又扱えるモデルの大きさも小さな地域モデルから世界モデルに至るまで千差万別であり、計量経済分析の立場から画一的に比較するのは困難である (これの展望については、森敬 [1974, 1977] に詳しく述べられている)。

電子計算機言語の立場からすれば、これらのシステムも、より利用者にとって馴みやすい言語体系のものから、電子計算機の機械言語に近いものまでいろいろあるが、本質的には翻訳 (コンパイラーまたはアセンブラー) 形式かインタプリタ (即時実行) 形式か、パラメータまたはメニュー形式か、又はこれらの組み合わせが何層にも渡っているかの違いだけである。このことに注目して、計量経済分析システムを新規に開発する場合の基本的な機能について

考えることにする。この様な言語体系間の差異は、汎用言語を含む電子計算機言語について一般的に言えることで、計量経済分析言語に特有ではない。さらに、分析の局面や対象機の性能・特性によって、利用者にとって使い易いものがよいので、どの方式が優れているとは、一概には言えない。

前節までの説明からも明らかな様に、計量経済分析のためのシステムを、汎用電子計算機上に作成する場合にも、これらを仮想電子計算機として捉えることが可能である。

これらの仮想電子計算機の違いは、その仮想電子計算機をどの層の上に更にどれだけの層を構成するかによって、またその構成方法を翻訳形式とするのかインタプリタ形式にするのか、あるいは、パラメータ形式かメニュー形式で処理するのか、によって分類することが出来る。

以降の章では具体的にこれらの計量経済システムの例について考察する。

## 第二章 計量経済分析システム

佐波隆光氏（「数量経済分析の基礎」）によれば、数量経済分析は「経験的に観測された数量をもちいておこなう経済分析」すなわち「ひろい意味での統計手法（記述統計手法もふくむ）の計量経済分析への応用の体系である」と定義される。さらに、狭義の計量経済分析は「経済理論によってあたえられる経済変数のあいだになりたつ諸関係式を、現実の統計データにもとづいて具体的に推定すること、さらに推定結果にもとづいて、予測をおこなったり政策の評価をおこなうこと」であるとされ、数量経済分析の一部として定義される。

つまり狭義の計量経済分析と数量経済分析との違いは、経済理論によって与えられる関係式の推定を、含むか含まないかという点にあり、記述統計的手法（平均、標準偏差、相関係数等）による分析、通常の時系列分析や分散分析などは、この定義の計量経済分析には含まれないことになる。

この佐波隆光氏の計量経済分析の定義を採用すれば、計量経済分析システムは、「経済理論によってあたえられる経済変数のあいだになりたつ諸関係式を、現実の統計データにもとづいて具体的に推定し、さらに推定結果にもとづいて、予測をおこなったり政策の評価をおこなう」システムといえることができる。

本章では、まずこのためのシステムの構成を考察し、さらに広く数量経済分析システムとするための機能についても考察することにする。

### 2.1 計量経済分析システムの機能

前章で説明した考え方を拡張すれば、計量経済分析のためのシステムも、仮想電子計算機で説明することが出来る。いろいろな方式の仮想電子計算機を想定することが出来るが、これらの、仮想電子計算機としての相違点は、

個々の仮想電子計算機のもつ層数と、その構成方法を翻訳形式とするのか、インタプリタ形式にするのか、またその組み合わせとするのか、によって分類することが出来る。また、計量経済分析システムを構成する場合、計量経済分析の分析手法の特殊性から逆にシステム構成が限定される場合もある。

このため本節では、計量経済分析をするためには計量経済分析システムに何が必要なのかをクラインの第一モデル [1950] を例に考察することにする。

### 2.1.1 計量経済モデルの構造の特化

計量経済モデルは、経済変数間の相互依存関係を連立方程式体系で表わしたものである。モデルに含まれる経済変数は大きく3つのグループに分類される。第一のグループは、内生変数と呼ばれる変数で、変数の値がモデルの相互依存関係の結果として決定される変数の集まりであり、第二のグループは、外生変数と呼ばれる変数で、変数の値がモデルの構成からは全く影響を受けず、モデルの外で決定されると仮定される変数の集まりである。最後に、第三のグループとして、先決内生変数と呼ばれるものがある。これらは内生変数の過去の値の集まりである。第二と第三のグループを併せて、そのモデルの先決変数と呼ぶ。ただし、経済モデルの場合、外生変数と言っても、内生変数に対して完全に独立であるとは限らず、モデルの構成にあたっての便宜上、外生として取り扱うものが多い。これらは統計的な意味では内生変数から独立ではなく、モデル内だけで、便宜上、独立とみなすに過ぎないので注意を要する。

つぎの手続きはこれらの変数を用いて、経済変数間の因果関係や定義を、方程式の形で記述することである。この記述は、電子計算機の演算式記法の慣用にしがって、等式の左辺に被決定変数名を書き、右辺に説明変数群とその係数とによって式の形で決定機構を記述する。この過程はモデル作成者の、経済理論的考察に基づく、対象とする事象についての経済観の数式的表明である。

こうして決定された個々の変数についての方程式による説明は、連立方程式にまとめあげられる。一つの式の被決定変数は、通常、他の（多くは複数の）

式の説明変数となる。こうして経済諸量間の相互依存関係が連立方程式によって説明されることになる。これらの被決定変数が、体系の内生変数である。

この様にして構成される連立方程式体系に含まれる方程式の数は、内生変数の数に等しい。各方程式はその経済学的意味づけから行動方程式、制度方程式、定義式（会計的恒等式）および統計式の何れかを表わしている。

当然、変形して整理すれば全く同じ経済的意味をもつようになる方程式が、複数個含まれる様な方程式体系であってはならない。

クラインの第一モデルを例にとり、計量経済モデルの連立方程式として定式化された姿を記述する。このモデルは1921年から1941年にかけてのデータによるアメリカ経済のマクロモデルであり、3本の行動方程式と5本の定義式から成る。その構造は次の様になっている。

- ・消費関数  $C = r_{11} + \beta_{14} P + r_{16} P_{-1} + \beta_{16} W$
- ・投資関数  $I = r_{21} + \beta_{24} P + r_{26} P_{-1} + r_{27} K_{-1}$
- ・民間賃金関数  $W^* = r_{31} + \beta_{38} E + r_{38} E_{-1} + r_{35} A$
- ・定義式  $P = Y - W$

$$Y = C + I + G - T$$

$$K = I - K_{-1}$$

$$W = W^* + W^{**}$$

$$E = Y + T - W^{**}$$

8本の方程式に対応した8変数、 $C$  = 消費、 $I$  = 投資、 $W^*$  = 民間賃金所得（総額）、 $E$  = 民間生産所得、が内生変数となっており、外生変数としては、 $W^{**}$  = 政府および公企業賃金所得、 $T$  = 間接税、 $G$  = 政府支出、 $A$  = タイムトレンド、の4変数と定数項がある。 $P_{-1}$ 、 $K_{-1}$ 、 $E_{-1}$ はそれぞれ対応する変数の前期の値、つまり先決内生変数である。外生変数の内、 $W^{**}$ 、 $T$ 、 $G$ は、政府によって決定されるものであり、その意味で外生であるが、必ずしも全経済量と独立には決定しえない。統計的な意味で



は独立変数ではないことに注意すべきである。

計量経済モデルの構造を、このようにして連立方程式の形で表現することを、モデルを「特化」(specify)するという。

### 2.1.2 データ整備と計量経済モデルの推定

計量経済モデルの定式化が終わると、次は各方程式の係数を推定しなければならない。言うまでもなく、これらの推定に必要な各変数についての十分な数の観測値が、推定に先だって、準備されていることが必要である。

観測値データの準備は計量経済モデル分析にとって相当のウェイトを占める作業である。大量のデータの入力・訂正・検認を比較的容易におこなうことが出来なければならない。その上、入力したデータを用いて、それを加工し、新しいデータを造り出す作業が、容易に可能でなければならない。さらに、これらのデータは、必要に応じて、ランダムに高速検索が可能でなければならない。データ個数が増大してくると、各々のデータの識別が次第に困難になってくる。個々のデータについての十分な説明が検索できるようにしておかないと、データの管理が重大な障害になってくる。

推定法としては不偏推定値を得るために各種の同時推定法が推奨されている。しかし、方程式が特に内生変数に関して非線形であったり、同時推定法で妥当な結果が得られないときなどには、直接(単純)最小二乗推定法が用いられ、推定式は一種の近似式として扱われる。方程式が特殊な条件をもつ場合には、それに対応して特殊な推定法が採用される場合もある。とにかく、同時推定法にしる、直接最小二乗推定法にしる、なんらかの方法で、C、I、W\*の各方程式を推定する。これらの推定法は、分析の過程で、必要に応じて任意に選択できる様になっていなければならない。

### 2.1.3 モデル・シミュレーション

与えられた計量経済モデルが、クラインモデルのような線形連立方程式モデルであれば、内生変数の係数行列の逆行列を連立方程式の両辺に乗ずることに

よって、各内生変数が外生変数と先決内生変数のみで表わされる関数式、つまり誘導型を導くことが出来る。そして、この誘導型に外生変数と先決内生変数のデータを代入すれば内生変数の値が求められる。しかし、計量経済モデルは一般には非線形連立方程式体系になってしまう場合が多く、この場合には、誘導型を求めて、それに外生変数と先決内生変数のデータを与え、連立方程式を解いて内生変数の値を求めることは、理論的に不可能である。

非線形モデルを解くには変形ガウス・ザイデル法と呼ばれる繰り返し計算法が用いられる。ガウス・ザイデル法は、まず非線形連立方程式体系の各方程式を一個の内生変数だけを左辺の（従属）変数とする式に書き換え、各内生変数が何れかの方程式の従属変数となるようにする。この書き換えられたモデルをシミュレーション・モデルと呼ぶことにしよう。クラインの第一モデルの場合は、前節のモデルをそのままシミュレーション・モデルとして用いることができる。なぜなら、モデルの各方程式が既に各内生変数一個だけを左辺の従属変数とする式として表わされているからである。このモデル自体は線形モデルである。しかし、線形モデルであっても変形ガウス・ザイデル法でモデルを解くことができる。

ガウス・ザイデル法は繰り返し計算法であり、その第一ラウンドはシミュレーション・モデルの式を定義した（プログラムされた）順に、式の右辺の変数に値を代入して従属変数の値を計算し、以下の式では内生変数の値としてはこの計算値を用いながら最後の式まで計算する。この第一ラウンドの計算で、内生変数の値として計算値が未だ利用出来ないときは、通常、一期前の観測値を初期値として用いる。第二ラウンドは、第一ラウンドで得た内生変数の計算値を利用しながら第一ラウンドと同様な計算を行なう。以上の計算を総ての内生変数の値が、一つ前のラウンドの計算値と比較して、ある割合の変化率（例えば1%）以下に収束するまで繰り返す。

この繰り返し計算法では内生変数の値は大低の場合収束するが、いつも必ず

収束するとは限らない。シミュレーション・モデルの式の並べ方が悪い場合、または、各ラウンドでの計算で前ラウンドで得た計算値をそのまま利用するため、ある条件になると循環して同じ値が得られる様になってしまう為に、収束しないこともある。これらの場合には式の順序を並べ変えたり、または、前のラウンドで得られた計算値の加重平均を新しい計算値としたりして、改善を試みる。

われわれの今までの経験では、非線形モデルの解法にあたって、特に言う程の大きな困難に遭ったことはない。

## 2.2 計量経済分析システムの構造

本節では、前節で述べた計量経済分析システムの機能を、システムを作成する立場から再検討してみることにする。

既に数多くの計量経済分析システムがあるが、主な差異は、モデルの解法に入るに先だって、必要なサブ・システム、例えば、データ・マネージメント、推定のために利用できる各種分析法等の、範囲、規模および利用者がプログラム・システムを使う時に与えねばならない各種のパラメータやデータの与え方にある。モデルの解法部分の設計の基本的構造については、殆ど差がないと言える。多くの解法システムは、次の様な構造になっている。

### 2.2.1 必要なデータの準備

解法プログラムを実行するには、次のデータが準備されていなければならない。これらの準備がプログラム・システムの中で行われるか、解法プログラムに対して独立に与えねばならないかは、われわれの問題ではない。

- (1) 各種パラメータ
- (2) 変数名表
- (3) 変数の値
- (4) モデルのスペシフィケーション

(1) パラメータには A) シミュレーションの期間 B) 繰り返し回数の制御 C) 収束判定条件 D) 変数に関する諸情報, 等が含まれる。その個数は多くて30個程である。

(2) 変数名表には, モデルに含まれている変数の名称が示される。内生変数, 外生変数を区別して与える場合と, 区別しないでシステム内部で方程式を解析して判別している場合とがある。これは, 方程式に含まれる変数名を計算機内部の変数に対応させて, 実行可能な解法プログラムを作り出す為に使用される。

(3) 変数の値はモデルに含まれる総ての変数についての既存の観測値, 推定値およびその他の値である。これらは解法の実行に当たって初期値または定数となる。外挿予測を行う場合は, 外挿期間についての外生変数の推定値を与えておかねばならない。

(4) モデルのスペシフィケーションには, 連立方程式モデルの内容が与えられる。プログラム・システムによって, Explicit に記述しなければならない場合と, Implicit な与え方の出来る場合とがある。

プログラム・システムの構造としては, まず, これらの予め準備された(1)–(4)のデータを読み込む。これらは, システムによって, 一括してカードその他, 何らかのタイプのファイルから与えられるか, または, 先行する推定作業等の過程で, 入力された情報から, 自動的に生成されて行くか, いろいろなタイプがある。

### 2.2.2 モデルのプログラムへの組み込み

利用者にとって最も大きい差異と写るのは, モデルの入力方法である。これには, システムによって,

- (1) FORTRAN, APL, BASIC 等の汎用プログラム言語を直接使う。
- (2) 専用言語を持ち, これを解析して, 他の汎用言語を生成する。

(3) 専用言語を解析して中間言語を作り、interpretative に実行する。等の方法がとられる。(1)の場合、APL, BASIC が使われれば、これらの言語はインタプリタ形式であって、効率的には(3)に近い。(2)の中にも、生成される汎用言語が、BASIC 等のインタプリタ言語である場合がある。

これらの殆どは、添数について若干の注意を必要とするだけで、変数名を使用して、モデルの教科書的な記法とあまり変わらない書き方で、モデルを示せばよい様に工夫され、プログラム・リストがモデル構造の説明に使える様になっている。しかし、汎用言語でモデルを記述するものの中には、readability に全く考慮を払わず、変数名のかわりに変数番号を用いることを要求しているシステムも、いまだに存在している。

一般的に言って、インタプリタ言語を用いたものは、モデルの修正、再実行が簡単で使い易いが、大きいモデルの場合、実行時間が長くなる難点がある。FORTRAN 等のコンパイラ言語を、直接、または、専用言語から生成して使用する方式は、翻訳、結合編集を必要とするので、修正に時間がかかるが、実計算時間は早い。今までの使用経験からすれば一長一短で、小型モデルには前者が向いており、大型モデルには後者がよい。その境界は方程式数にして100本程度であるといえる。最近では、場合に応じてこの両者を選択して使用できるシステムも出現している。

モデル構造の記述部分は、必要ならば解析を行なった後、本体となるプログラムの連結部分を付加する等の処理を行ない、ガウス・ザイデル解法が実行可能な形のプログラムに編集される。この場合、一定の基準に従って方程式の順序を並べ変えるものもある。

この様にして必要な処理が終わると、解法プログラムは、必要ならば翻訳されて、プログラム本体に、結合編集される。この部分を SUBROUTINE SOLVE と呼ぶことにしよう。SOLVE の取り込みが終わると解法

(simulation) が実行される。

### 2.2.3 モデルの解法

モデル解法部分のプログラムについて、概念的な流れ図を第2-1図に示した。システムは、次の5つの部分から成る。

- (1) Simulation 期間のカウント
- (2) Iteration 回数制御
- (3) 1期間の解法 (CALL SOLVE)
- (4) 収束結果の判定
- (5) Simulation の終了処理

更に、Simulation が発散その他の異常終了をしなかった場合には、

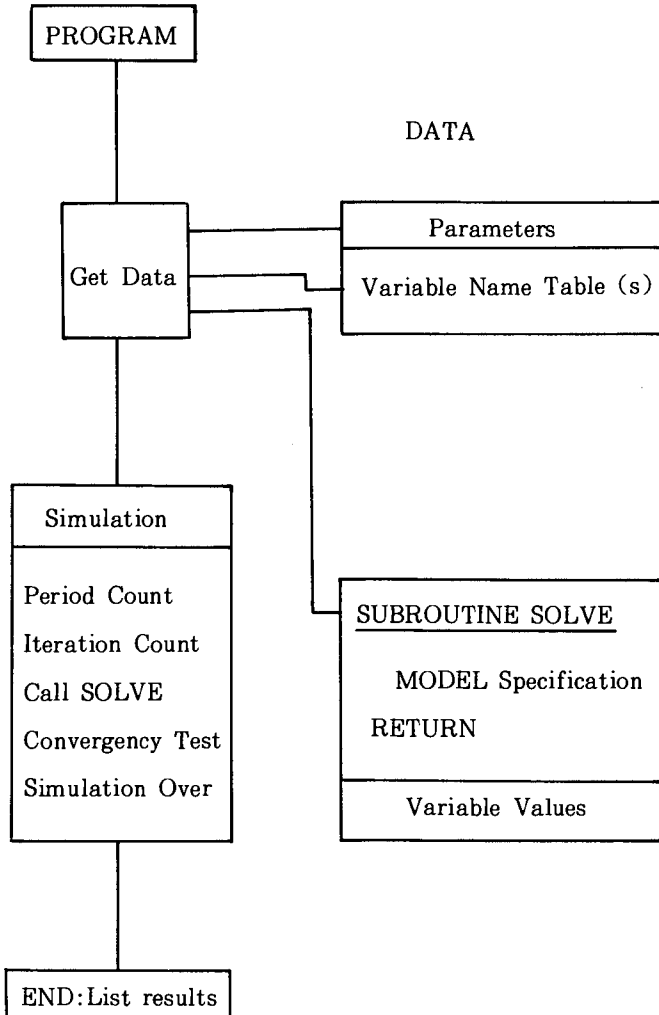
- (6) Simulation 結果の Output

が実行される。

細部について言えば、user friendly で、モデル分析を容易にするための種々の利用者支援用プログラムを組み込んであるものから、利用者に対する配慮を欠いていて、そのシステムに慣れたものでなければ、とても使えない様なものまで、様々であるが、基本的には、計量経済モデル解法プログラム・システムは以上の様な構造をしていると言える。

以上の説明からも分かる様に、計量経済分析システムは前章で説明した仮想電子計算機に相当することが理解できる。つまりその構成方法がインタプリタ形式か、翻訳形式か、またその組み合わせなのかが違うだけである。しかしモデル解法についてだけ考えると、その特殊性、つまりモデルの一行ごとを一命令と考えるとモデル全体の命令が、前もって与えられなければ解くことが出来ないため、一括処理形式を取らねばならない。そのため翻訳形式（一部翻訳形式のシステムも含める）を採用しているシステムが多い。細部については次章以降に述べる。

第2-1図 単一モデルシステム



### 第三章 計量経済分析システムの具体例

世界の各機関で開発されている代表的な計量経済分析システムの多くは、FORTRAN言語（一部PL/I言語で記述されているシステムもある）で記述され、パラメータ方式から独自の高度な言語体系を持つものまで千差万別である。パラメータ方式も広義に解釈すれば言語とみなすことが出来るし、最近は分散処理端末のインテリジェンシィの発展にともなって、メニュー方式による対話型処理が一般化する傾向にある。メニュー方式はパラメータ方式の一種であるから、本文ではパラメータ方式も電子計算機言語として統一的に扱うことにする。

これらの中から言語体系からみて代表的な計量経済分析システムの例として、パラメータ方式のペンシルバニア大学のシステム、翻訳方式とインタプリタ方式を組み合わせた TSP（Time Series Processor）と STEPS（Statistical Techniques for Econometric Planning and Simulation）、そしてインタプリタ方式の STS（Stochastic Simulation System）IAS（Inter-active Simulation System）を取り上げ前章の定義に従って具体的な例と共に仮想電子計算機として分類し、各々のシステムの特徴を考えてみる事にする。

またペンシルバニア大学のパラメータ方式のシステムは、パラメータ方式というだけではなく、現在でも多方面で利用されている手法が多く採用されている。このためペンシルバニア大学のシステムについては、実際にモデルを解くために採用された手法についても述べることにする。



### 3.1 ペンシルバニア大学のシステム

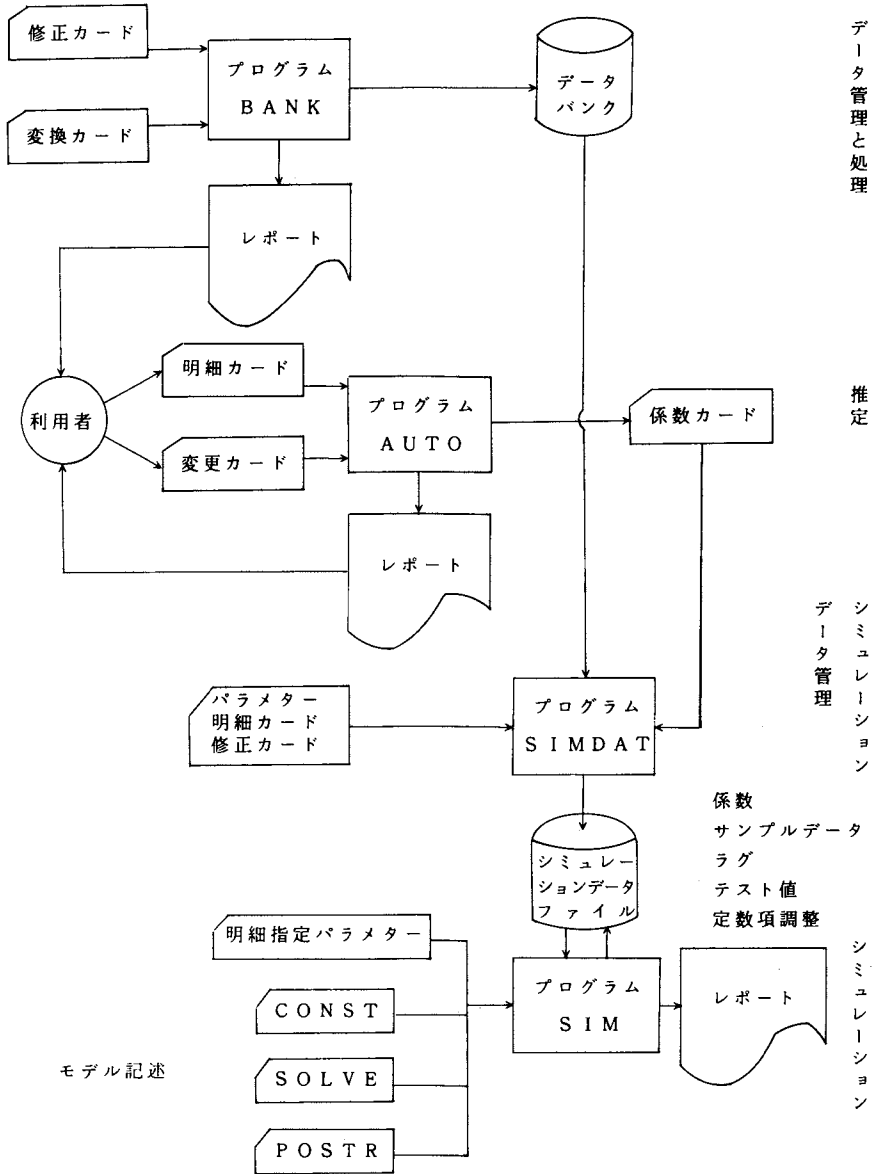
第3-1図にペンシルバニア大学で1960年代後半から1970年代にかけて開発された計量経済分析システムの全体の構成図を示す。このモデル記述部分は、線形でも非線形でも解くことが可能であり、基本的部分については WEFA (Wharton Econometric Forecast Associate) でも使用され、FORT RAN でモデルを記述するシステムとしては世界的に利用されたシステムである。このシステムはプロジェクト・リンクのために幾つかの機能が追加されており(第3-2図参照)、開発当時の電子計算機の大きさでもプロジェクト・リンクで使用される様な大型モデルが分析出来る様に随所に当時の最新の手法が採用されていた。

これらの手法の多くは、最近の計量経済分析システムでもそのまま採用されていることが多く、当時としては画期的なことであった。また大型モデルを解く場合、計算時間が他の方法に比べて最も速いという理由で最近でも多く採用されている。このため以後の計量経済分析システムを考える上で本システムの構造は参考になると思われるので、その詳細についてクラインの第一モデルを例として述べることにする。

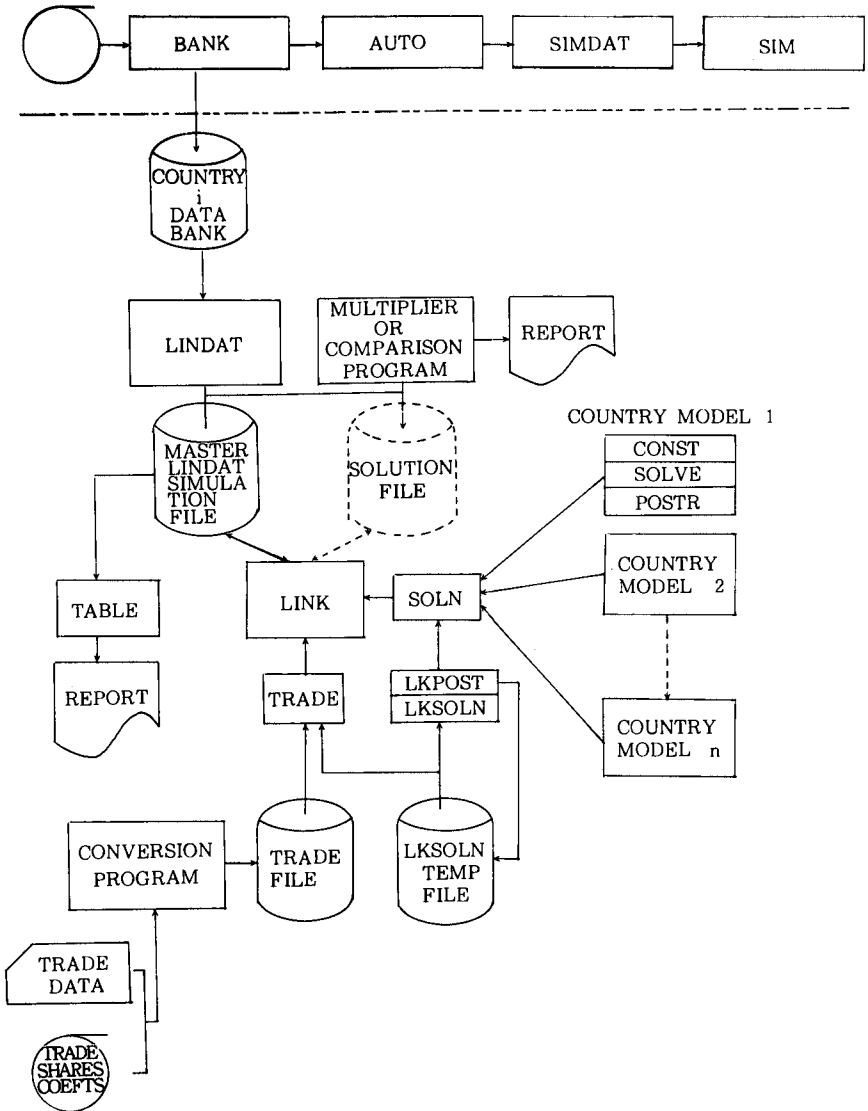
この計量経済分析システムが開発された時代には、まだパッケージプログラムという考え方が定着していない時代であり、そのため独自の言語体系をシステムが持つということは行なわれなかった。その代わりパラメータ方式と呼ばれる方法が採用される事が多かった。このペンシルバニア大学のシステムも、パラメータ方式を採用して構成されたシステムである。パラメータを、ある種の簡易言語の初期の形態とみなすことも出来るので、以下本文では、パラメータ方式もある種の言語体系として扱うことにする。それ故このシステムも仮想電子計算機として分類することにする。

この計量経済分析システムは一つのシステムから出来上がっているのではなく、

第3-1図 ペンシルバニア大学のシステム



第3-2図 ペンシルバニア大学のプロジェクト・リンクのシステム



- (1) BANK
- (2) AUTO
- (3) SIMDAT
- (4) SIM

の4つの独立のサブシステムで構成されたシステムである。更にプロジェクト・リンクのために

- (5) LINDAT
- (6) LINK

の2つのサブシステムが追加されている。各々のシステムを有機的に繋ぐ方法として、当時最も一般的であったファイルによる方法が用いられていた。これら6つの単一システムの仮想電子計算機を考えてみる。

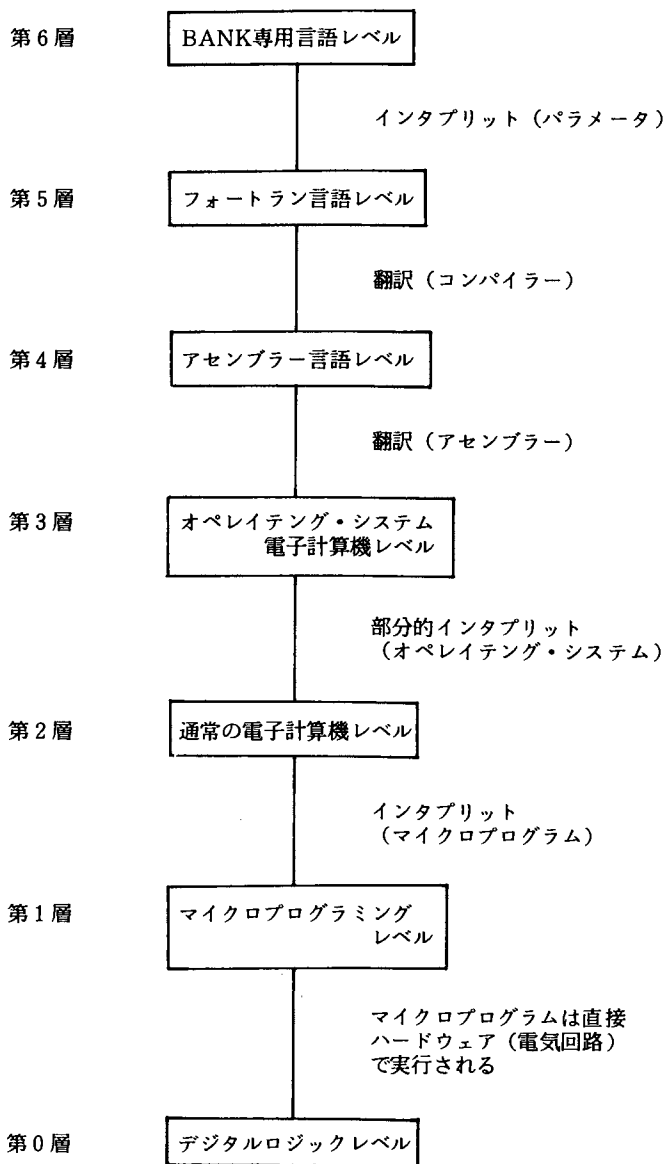
#### 3.1.1 BANKシステム

このBANKシステムは、次のAUTOシステムとSIMDATシステムのためのデータ・ファイル（このシステムではデータ・バンクという呼び方を採用している）を作成するためのシステムであり、第3-3図の様な仮想電子計算機が考えられる。追加・修正と加工のためのパラメータと、これに付随するデータを一種の機械語とする第6層目の仮想電子計算機が、この機械語をインタプリットして実行される。このBANKシステム自身はFORTRAN言語で記述されており、第5層目以下は第1-2図と同様になる。

このシステムの特徴は、データをAUTOシステムやSIMDATシステム等に使用する際、データを任意に読み出すことが可能である様に構成されている。そのためFORTRANによる直接アクセス入出力命令でランダムにデータを取り出せるように設計されている。

パラメーター形式のシステムを説明する材料として、このシステムとシミュレーション用のシステムについては詳細な説明を与えておく。

第3-3図 BANKシステムの仮想電子計算機



第1番目のカード 国名カード CNAME, ABV  
(A 8, 2 X, A 4)

カラム

1-8 CHAME 国名

11-14 ABV 国名の省略名

第2番目のカード パラメーター カード

NV, N1, N2, N3, NO1,  
IY2, IP2, IB2, N4, N5,  
(16I5)

カラム

1-5	NV	= 0	N5で指定されたオプションによる 通常のデータファイル作成
		≠ 0	第三番目のカードのセットによるデ ータファイルの修正, または標準形 式によるデータファイルの作成
6-10	N1	= 0	副プログラムUPTRANを呼ばな い
		≠ 0	副プログラムUPTRANを呼ぶ
11-15	N2	= 0	標準
		= n	nは100以下の整数で, 作表する系 列の数を表わす。作表する系列の指 定は第15番目のカードで指定する。
		= 101	全系列の作表を行なう
		= 102	全系列名と系列番号の作表を行なう
		= 103	全系列名のアルファベット順の作表 を行なう

16-20	N 3	= 0	旧データファイルの修正とその修正表の作成
		= 1	新データファイルの作成
		= 2	N 5 のオプションで指定された入力形式のカードによるデータファイルの再構築
21-25	NO 1	=	作表すべき観測値の数
29-30	IY 2	=	データの開始年(西暦年数の下2桁)
34-35	IP 2	=	一年あたりのデータの数(期種の指定)
39-40	IB 2	=	作表の開始期
45	N 4	= 0	IP 2 によって指定された期種とデータの期種が違う場合, 作表しない
		= 1	期種がデータと違っていても全てデータ期種に合わせて作表する
48-50	N 5	= 0	標準
		=	カードにパンチする系列の数 パンチすべき系列は第16番目のカードで指定する。
第3番目のカード	修正カード		NAME, NUM, NT, IY, IP, IB, NC, NO, NEW, Y (A 8, I 4, I 1, I 2, 5 I 3, 5 F10.5)
カラム			
1-8	NAME	=	系列名(左詰め)
		= \$\$\$	第7番目のカードによるデータの入

			力（左詰め）
9-12	NUM	=	データファイル上での論理レコード の番号
13	NT	= 0	単一カードによる修正
		= 1	標準カードによる修正
		= 2	書式指定による修正
14-15	IY	=	データの開始年度（西暦年数の下2 桁）
16-18	IP	=	一年あたりのデータの数（季種の指 定） もし空白の場合IP2の値が使用さ れる
19-21	IB	=	開始期の指定 もし空白の場合IB2の値が使用さ れる
22-24	NC	=	標準カード修正の場合のカード一枚 あたりのデータの数 省略時の値は、年データの場合5 その他の場合は4
25-27	NO	=	修正すべきデータの数（単一カード 修正の場合は指定しない）
28-30	NEW	= 0	旧系列の修正
		= 888	旧系列を破棄して新系列を作成
		= 999	新系列を追加
31-80	Vector Y	=	単一カード修正の場合の修正値 （最大5個まで）



## 第4番目のカード 標準データカード

(7A4, A2, 5F10.5)

このカードは第3番目のカードのNT=1の場合のみ必要である。その他の場合このカードは指定しない。このカードの書式は第3番目のカードの書式と同一である。但し最初の30桁は無視される。

## 第5番目のカード 書式指定によるデータの修正の場合の書式

(20A4)

第6番目のカードで読み込まれるデータの書式を指定する。例えば(12F6.3)の様に指定する。

## 第6番目のカード 書式指定の場合のデータカード

第5番目のカードで指定された書式に従って入力されるデータ。

## 第7番目のカード 修正パラメーターカード

NL, NO, M1, MN, M2, M3,  
IY, IP, IB, NV, M4,  
(11I5)

## コラム

1-5	NL	=	Zマトリックスの列方向の開始列の値, 初期値は1
6-10	NO	=	Zマトリックスの列方向の終了列の値, 初期値は1
16-20	MN	=	Zマトリックスの行方向の終了行の値
25	M2	= 0	第8番のカードで示されるレコードの値でZマトリックスを埋める
		= 1	第9番目のカードで示されるFORMATに従って観測値を読む

		= 2	第 9 番目のカードで示される FOR MATに従ってデータを読む
30	M 3	= 0	標準
		= 1	Zマトリックスにデータを格納した 後、UPTRAN用のパラメータを 読む
		= 2	データ変換ならびにDATRAN処 理が終わった後UPTRANパラメ ータを読む
34-35	IY	=	Zマトリックスの開始年度（西暦年 数の下 2 桁）
39-40	IP	=	一年あたりのデータの数（季種の指 定）もし空白の場合IP2の値が使 用される
44-45	IB	=	開始期の指定  もし空白の場合IB2の値が使用さ れる
46-50	NV	= 0	標準
		=	データバンクに記録するデータの数
55	M 4	= 0	標準
		=	DATARNを呼ぶ回数
第 8 番目のカード			ディスク上での位置指定カード  NUM, NAME  (I4, 1X, A8)

M 2 が 0 の場合のみ読まれる

## カラム

- 1-4 NUM = 変数の論理レコードの位置  
 6-13 NAME = データバンク上での名前  
 第9番目のカード 書式指定によるデータ入力の場合の書式  
 (20A4)

M2が1又は2の時のみ読まれる。この書式に従って、第10番又は第11番のカードが読まれる

- 第10番目のカード データカード  
 M2が1の時のみ読まれる

- 第11番目のカード 各々のデータのデータ名  
 (8(A8, 2X))

M2の値が0より大きい場合のみ読まれる

- 第12番目のカード DATRANプログラムに対するパラメータカード  
 DATRANプログラムから読まれる場合、この位置に必要なカードを置く

- 第13番目のカード 追加データのデータ名  
 (8(A8, 2X))

DATRANによってデータの系列数が増加した場合の増加した変数に対するデータの名前

- 第14番目のカード 転送カード NAME, NUM, IC, NL, NO,  
 NEW, IY, IP, IB  
 (A8, 2X, 8I5)

## カラム

- 1-8 NAME = 変数の名前  
 11-15 NUM = 変数の論理レコード番号  
 16-20 IC = 格納すべき変数のデータマトリック

			ス上での桁数
21-25	NL	=	IY年IB期の列方向の位置
26-30	NO	=	データの列方向の最後の位置
31-35	NEW	= 0	既にデータバンク上にデータがある 場合指定する
		= 888	旧データに代えて新しいデータと置 き換える
		= 999	新規に新しいデータを作る
39-40	IY	=	開始年度（西暦年数の下2桁）
44-45	IP	=	一年あたりのデータの数（季種の指 定）もし空白の場合IP 2 の値が使 用される
49-50	IB	=	開始期の指定 もし空白の場合IB 2 の値が使用さ れる
第15番目のカード	データリスト作成		
			(16I 5)

N 2 の指定に従ってリストすべき変数の番号を指定する

第16番目のカード	データパンチ作成		
			(16I 5)

N 5 の指定に従ってパンチすべき変数の番号を指定する

また、このパラメータ以外に本システムは、第2番目のカードの指定に従いUPTRAN, DATRANのFORTRANサブ・プログラムが必要である。第3-4図にDATRANのFORTRANサブルーティンの例を示す。

### 3.1.2 AUTOシステム

AUTOシステムは、このシステムに対するパラメータに従って前節のBA

## 第3-4図 DATRANの例

```

SUBROUTINE DATRAN (M4)
COMMON NS, NS1, MAXS, N1, N2, NO1, IY2, IP2,
      IB2, MAXC, NL, NO, MN, N4,
2      MAXNS, N6, M1,
3      /DATA/Z(100,50), LABEL(50), LOC(50)
DOUBLE PRECISION LABEL
M1 = 3
MN = 4
DO 1 I = 1, NO
  Z(I, 3) = Z(I, 3) * Z(I, 5)
1 Z(I, 4) = ALOG(Z(I, 4))
RETURN
END

```

NKシステムで作成されたデータ・ファイルから必要なデータを抽出し、場合によっては加工を行ない回帰分析を行なう。つまり前節と同様にパラメータを機械語として回帰分析をインタプリタ形式で実行することになる。この場合の仮想電子計算機を第3-5図に示す。

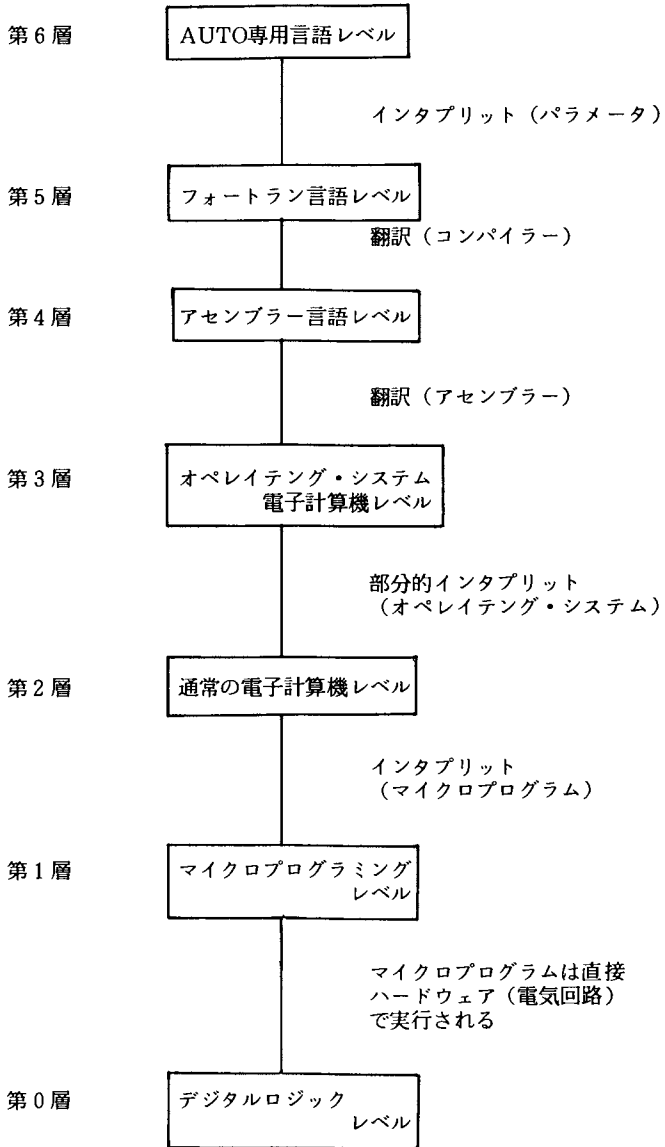
## 3.1.3 SIMDATシステム

このシステムもパラメータに従ってBANKシステムで作成されたデータ・ファイルを次のSIMシステムのためのデータファイルに変換するシステムである。このシステムも前節と同様にインタプリタ形式で実行する仮想電子計算機を考える事が出来る。この場合の仮想電子計算機を第3-6図に示す。

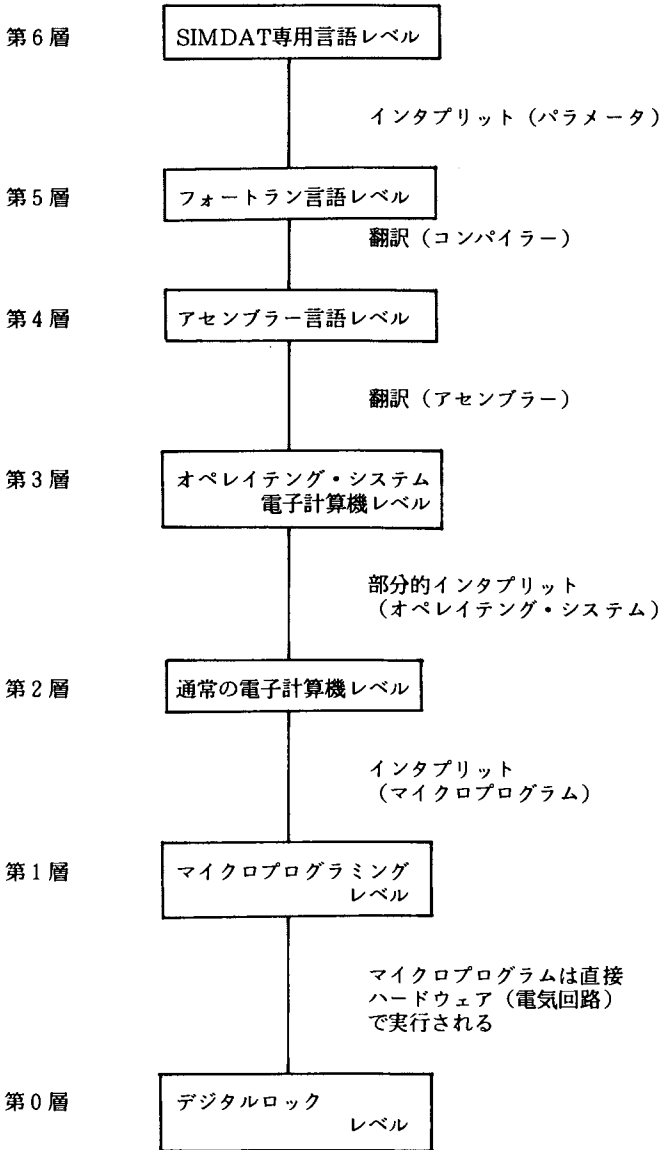
## 3.1.4 SIMシステム

このシステムは、これ自身では未完成であり、上記した3種類のシステムとは異なっている。つまり第6層目の仮想電子計算機そのものを利用者自身が作

第3-5図 AUTOシステムの仮想電子計算機



第3-6図 SIMDATシステムの仮想電子計算機



らなければならない。このために利用者が作成しなければならない部分は、第3-1図に示したCONST, SOLVE, POSTRに相当する部分であり、これらをFORTRAN言語で記述し、その後コンパイル、リンケージによって仮想電子計算機を作成しなければならない。

こうして完成した仮想電子計算機は、上記した三種類のシステムと同様にパラメータを機械語としてインタプリタ形式で実行されるシステムとなる。言い換えると、ある特定のモデルを計算する仮想電子計算機が作成されることになる。

この構成要素で最も重要な部分であるモデルの解法は、ガウス・ザイデル法を用いている。つまり外生変数に対して確定した値を与え、内生変数に対してある初期値（以後の計算過程でこの変数値は変化する）を与えた後、モデルの記述（正確にはプログラムの記述）に従って順次式を計算してゆき最後の方程式迄計算が終われば、その時の内生変数の値を初期値として再度始めから順次式を計算するという手順を何度か繰り返す。この時最後の式の計算が終わった時点で、前回の計算の時の各々の内生変数の値と今回の各々の内生変数の値とを比較してこの各々の内生変数の値の変化（変化率）がある値以下であれば解が求まったとする。

この計算方法をよく見ると次の様な三つの部分に分けることが出来る。第一に外生変数の値が決まればこの繰り返し計算の途中で変化が見られない内生変数を決定する式、第二に繰り返し計算の途中で毎回値が変化する内生変数の式であるが、この値が他の内生変数の式に影響を与えない式、そして第三に毎回変化し、かつ他の内生変数の式に影響を与える式の三つである。

このシステムでは、これらを三つに分解してプログラミングする方法が採用されている。第一番目をモデルの先決部分と呼び、本システムではCONSTと呼ばれるプログラムとして記述する。第二番目をモデルの後決部分と呼び本システムではPOSTRと呼ばれるプログラムとして記述する。残りの部分を再帰部分と呼び、このシステムではSOLVEと呼ばれるプログラムで記述する。これら三つの部分が組み合さって、一つのモデルを解く部分を構成する。



以下本文では、これら三つの部分をまとめてSOLVEと呼ぶことにする。

このシステムは未完成ではあるが、もしなんらかの方法でこの部分が作成され完成すれば、これ自身がある特定のモデルを解く仮想電子計算機となる。つまりこの処理が終われば、システムは初期条件、打ち切り条件等をこの仮想電子計算機の機械語として処理しうる。これは前節の仮想電子計算機と本質的には同じものである。

このため仮想電子計算機を完成させるための処理が必要になってくる。この様子を第3-7図に、クラインの第一モデルの場合の例を第3-8図に示す。

### 3.1.5 LINDATシステム

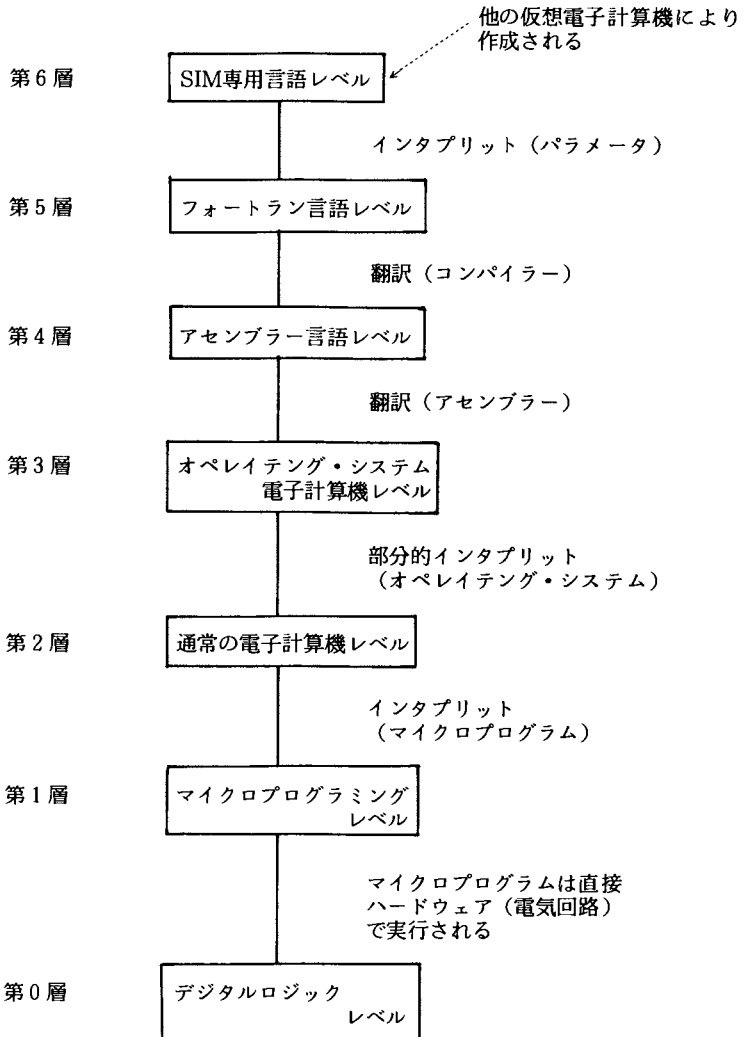
このシステムは、パラメータに従ってSIMDATシステムで作成された各国のデータファイルを併合して次のLINKシステムのためのデータファイルに変換するシステムである。このシステムも前節と同様に、インタプリタ形式で実行する仮想電子計算機を考える事が出来る。この場合の仮想電子計算機を第3-9図に示す。

### 3.1.6 LINKシステム

このシステムは基本的には3.1.4節のSIMと同じで、ただ各国のモデルが順次実行される様に制御する部分が追加されているだけである。つまり世界モデルを解く仮想電子計算機が実在することになる。パラメータを機械語として世界モデル仮想電子計算機が実行されるにすぎない。世界モデルを解くための、各国のモデルの実行順序等の制御については第4章に詳しく述べる。

しかし、この方法は第3-8図の例からも分かる様に変数や回帰分析した係数等を添数番号で記述するため、モデルの解読が困難であるばかりでなく、モデルの修正や変更等を行なう場合人為的間違いを発生しやすくしている。これに対して、このシステム以後開発されるに至ったシステムでは、計量経済分析をする者にとって馴染みやすい変数名（GNP, GDP等）を直接使用する様にして上記した欠点を除去しているが、大型モデルの場合、この繁雑さをして

第3-7図 SIMシステムの仮想電子計算機



## 第3-8図 クライン・モデルの例

```

SUBROUTINE CONST (.....)
      :
C      T-WG
      C(1)=Z(5)-Z(0)
C      1.497+0.146 * E-1+0.1302 * TIME
      C(2)=A(1)+A(2) * XL(1)+A(3) * Z(1)
C      10.1257+0.333 * P -0.1117 * K
      C(4)=A(4)+A(5) * XL(6) * XL(8)
C      G-T
      C(5)=Z(12)-Z(9)
C      16.2365+0.8988 * P
      C(7)=A(7)+A(8) * XL(6)
      RETURN
      END
SUBROUTINE SOLVE (.....)
      :
C      E=Y+T-XG
      Y(1)=Z(5)+C(1)
C      WP=1.497+0.4394 * E+0.146 * E-1+0.1302 * TIME
      Y(2)=A(9) * Z(1)+C(2)
C      W=WP+WG
      Y(3)=Z(2)+Z(0)
C      I=10.1257+0.4796 * P+0.333 * P-1-0.1117 * K-1
      Y(4)=A(0)+C(4)
C      Y=C+I+G-T
      Y(5)=Z(7)+Z(4)+C(5)
C      P=Y-W
      Y(6)=Z(5)-Z(3)
C      C=16.2365+0.1929 * P+0.08988 * P-1+0.796 * W
      Y(7)=A(8) * Z(6)+C(7)
      RETURN
      END
SUBROUTINE POSTR (.....)
      :
C      K=K-1+I
      Y(8)=XL(8)+Z(4)
      RETURN
      END

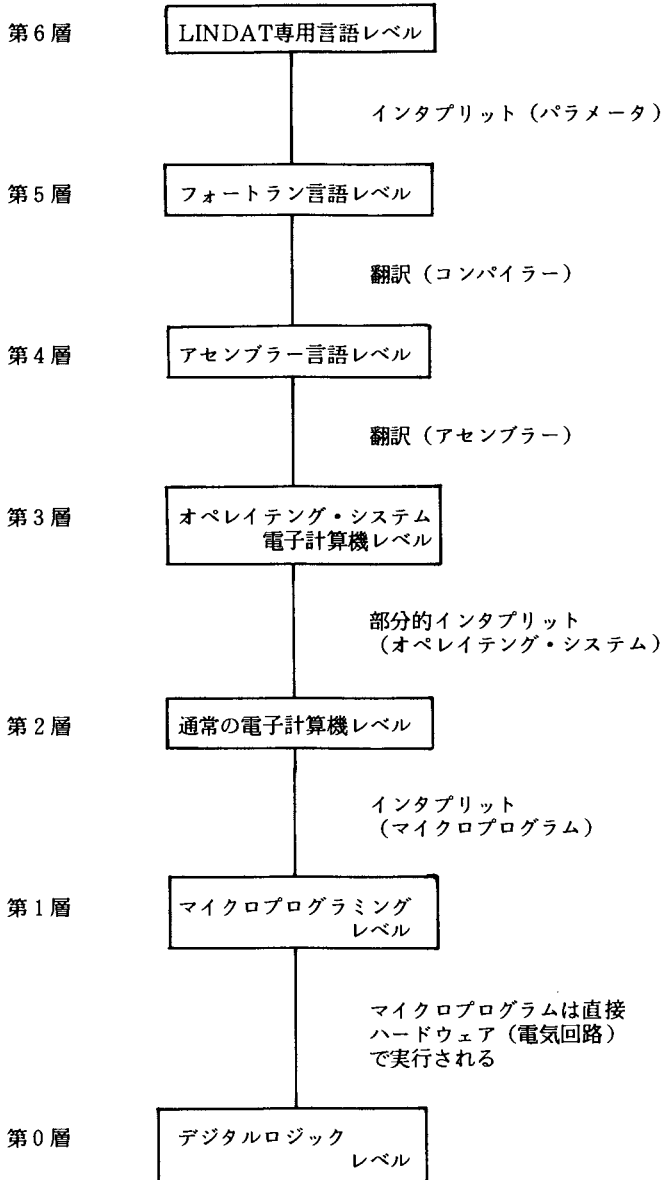
```

A (n) : 各式の係数  
C (n) : n番目の式の先決部分の値  
XL (n) : n番目の変数の一期ラグの値  
Y (n) : 各式の結果の代入先  
Z (n) : n番目の変数の値

添字と変数の対応表

1 : E	2 : WP	3 : W	4 : I	5 : Y
6 : P	7 : C	8 : K	9 : T	10 : WG
11 : TIME	12 : G			

第3-9図 LINDATシステムの仮想電子計算機



も計算時間の速さ（計算コスト）には替えられなく、現在もこのFORTRANによる方法が多く行なわれている。これらの欠点を除去するため次節以降で述べるシステムとこのFORTRANによる方法とを場合により使い分ける方法も色々考えられている。この一つの方法にモデルの構築中は後述するシステムを使用し、その後FORTRANによるモデル記述に書き換えを機械的に行なう、プレコンパイラーによる方法もある。この例として筆者によるペンシルバニア大学の汎用FORTRANモデルゼネレーターがあるが、これについては別稿に譲ることとする。

### 3.2 計量経済分析言語

前節で説明したパラメータ方式のシステムは、電子計算機の専門的知識のない利用者にとっては非常に複雑なシステムである。これに対して電子計算機の専門的知識がなくても計量経済分析が行なえることを目的として開発されたシステムについて述べる。つまり計量経済分析をする者にとって、彼らが通常使用している記述を、その機械語とする仮想電子計算機である。その代表的な翻訳方式とインタープリタ方式の例について示す。

#### 3.2.1 TSPとSTEPSシステム

TSP（Time Series Processor）とSTEPS（Statistical Techniques for Econometric Planning and Simulation）のサブシステム“MODEL”は、言語体系からすれば同一の構造を持っている。TSPもSTEPSも各々独自の言語（命令）体系を持ち、これらの言語（命令）を独自の中間言語に翻訳（コンパイル）し、その後この中間言語をインタープリットすることによって実行される。

つまり5層の仮想電子計算機の上に、さらに2層の仮想電子計算機が構築されていることになる。第6層目の仮想電子計算機は中間言語を機械語とする電子計算機であり、第7層目の仮想電子計算機はTSPまたはSTEPS言語（命令）

を機械語とする電子計算機となる。第7層目から第6層目の仮想電子計算機への制御の受け渡しは翻訳（コンパイラ）形式で行なわれ、第6層目から第5層目の仮想電子計算機への制御の受け渡しはインタプリタ形式で行なわれる。

この例としてTSPによるクラインの第一モデルの例を第3-10図に示す。

### 3.2.2 STSとIASシステム

STS（Stochastic Simulation System）とIAS（Inter-active Simulation System）システムとは、言語形態からすると同一の構造を持っている。つまりどちらも独自の言語（命令）体系を持ち、この言語（命令）を機械語としてインタプリタ形式で実行される。つまり5層の仮想電子計算機の上に、更にもう一つの仮想電子計算機が構築されていることになる。第6層目の仮想電子計算機はSTSまたはIAS言語（命令）を機械語とする電子計算機となる。第6層目から第5層目の仮想電子計算機への制御の受け渡しはインタプリタ形式で行なわれる。ただし前章の説明からも分かる様に全てがインタプリタ形式で実行されるのではなく一部分についてはコンパイラ形式の形態をとっている。しかし使用者からすればこの部分については見えなくて、すべてがインタプリタ形式で実行されているように見える。

クラインの第一モデルのSTSによる例を第3-11図に、また結果を付録に示す。

以上代表的な計量経済分析システムについて見て来たが、これらについて共通していえることは、データの入力・加工、推定そしてモデルの解法と三つの独立した仮想電子計算機がその時々に応じて順次（シリアル）実行されることである。システムによってはこれらの仮想電子計算機の変更を利用者からは分からない様に切り換えているものもある。この場合は、この切り換えに使用している部分についても仮想電子計算機と捉えることが可能である。

## 第3-10図 TSPによるクライン・モデル

```

NAME KLEIN;
LOAD;
SMPL 1 22;
GENR W=WP+WG;
GENR Y=CN+I+G-T;
GENR E=Y+T-WG;

SMPL 2 22;

PARAM A1, A2, A3, A4;
FRML CN0, CN=A1+A2*P+A3*P(-1)+A4*W;
LSQ CN0;

PARAM A5, A6, A7, A8;
FRML I0, I=A5+A6*P+A7*P(-1)+A8*K(-1);
LSQ I0;

PARAM A9, A10, A11, A12;
FRML WP0, WP=A9+A10*E+A11*E(-1)+A12*TIME;
LSQ WP0;

IDENT Y0, Y=CN+I+G-T;
IDENT P0, P=Y-W;
IDENT W0, W=WP+WG;
IDENT E0, E=Y+T-WG;
IDENT K0, K=K(-1)+I;

LIST ENDOGL CN I WP Y P W E K;
LIST EQU L CN0 I0 WP0 Y0 P0 W0 E0 K0;
MODEL EQU L, ENDOGL, KLEIN;

SOLVE (DYNAM) KLEIN;

STOP;
END;
SMPL 1 22;
LOAD CN;
39.8 41.9 45.0 49.2 50.6 52.6 55.1 56.2 57.3 57.8 55.0 50.9 45.6 46.5 48.7
51.3 57.7 58.7 57.5 61.6 65.0 69.7
LOAD I;
2.7 -0.2 1.9 5.2 3.0 5.1 5.6 4.2 3.0 5.1 1.0 -3.4 -6.2 -5.1 -3.0 -1.3
2.1 2.0 -1.9 1.3 3.3 4.9
LOAD WP;
28.8 25.5 29.3 34.1 33.9 35.4 37.4 37.9 39.2 41.3 37.9 34.5 29.0 28.5 30.6

```

## 第3-10図 TSPによるクライン・モデル

33.2 36.8 41.0 38.2 41.6 45.0 53.3

LOAD P;

12.7 12.4 16.9 18.4 19.4 20.1 19.6 19.8 21.1 21.7 15.6 11.4 7.0 11.2 12.3 14.0

17.6 17.3 15.3 19.0 21.1 23.5

LOAD K;

182.8 182.6 184.5 189.7 192.7 197.8 203.4 207.6 210.6 215.7 216.7 213.3 207.1

202.0 199.0 197.7 199.8 201.8 199.9 201.2 204.5 209.4

LOAD TIME;

-11. -10. -9. -8. -7. -6. -5. -4. -3. -2. -1. 0. 1. 2.

3. 4. 5. 6. 7. 8. 9. 10.

LOAD WG;

2.2 2.7 2.9 2.9 3.1 3.2 3.3 3.6 3.7 4.0 4.2 4.8 5.3 5.6 6.0 6.1 7.4 6.7 7.7

7.8 8.0 8.5

LOAD X;

44.9 45.6 50.1 57.2 57.1 61.0 64.0 64.4 64.5 67.0 61.2 53.4 44.3 45.1 49.7

54.4 62.7 65.0 60.9 69.5 75.7 88.4

LOAD G;

4.6 6.6 6.1 5.7 6.6 6.5 6.6 7.6 7.9 8.1 9.4 10.7 10.2 9.3 10.0 10.5 10.3 11.0

13.0 14.4 15.4 22.3

LOAD T;

3.4 7.7 3.9 4.7 3.8 5.5 7.0 6.7 4.2 4.0 7.7 7.5 8.3 5.4 6.8 7.2 8.3 6.7 7.4

8.9 9.6 11.6

END;



## 第3-11図 STSによるクライン・モデル

```

* FILE,I KLEIN
* DATA,I C * 1920 * 1941
39.8 41.9 45.0 49.2 50.6 52.6 55.1 56.2 57.3 57.8 55.0 50.9 45.6 46.5 48.7
51.3 57.7 58.7 57.5 61.6 65.0 69.7
* DATA,I I * 1920 * 1941
2.7 -0.2 1.9 5.2 3.0 5.1 5.6 4.2 3.0 5.1 1.0 -3.4 -6.2 -5.1 -3.0 -1.3
2.1 2.0 -1.9 1.3 3.3 4.9
* DATA,I WP * 1920 * 1941
28.8 25.5 29.3 34.1 33.9 35.4 37.4 37.9 39.2 41.3 37.9 34.5 29.0 28.5 30.6
33.2 36.8 41.0 38.2 41.6 45.0 53.3
* DATA,I P * 1920 * 1941
12.7 12.4 16.9 18.4 19.4 20.1 19.6 19.8 21.1 21.7 15.6 11.4 7.0 11.2 12.3 14.0
17.6 17.3 15.3 19.0 21.1 23.5
* DATA,I K * 1920 * 1941
182.8 182.6 184.5 189.7 192.7 197.8 203.4 207.6 210.6 215.7 216.7 213.3 207.1
202.0 199.0 197.7 199.8 201.8 199.9 201.2 204.5 209.4
* DATA,I WG * 1920 * 1941
2.2 2.7 2.9 2.9 3.1 3.2 3.3 3.6 3.7 4.0 4.2 4.8 5.3 5.6 6.0 6.1 7.4 6.7 7.7
7.8 8.0 8.5
* DATA,I G * 1920 * 1941
4.6 6.6 6.1 5.7 6.6 6.5 6.6 7.6 7.9 8.1 9.4 10.7 10.2 9.3 10.0 10.5 10.3 11.0
13.0 14.4 15.4 22.3
* DATA,I T * 1920 * 1941
3.4 7.7 3.9 4.7 3.8 5.5 7.0 6.7 4.2 4.0 7.7 7.5 8.3 5.4 6.8 7.2 8.3 6.7 7.4
8.9 9.6 11.6
* DATA,I TIME * 1920 * 1941
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4
5 6 7 8 9 10
* DATA,I X * 1920 * 1941
44.9 45.6 50.1 57.2 57.1 61.0 64.0 64.4 64.5 67.0 61.2 53.4 44.3 45.1 49.7
54.4 62.7 65.0 60.9 69.5 75.7 88.4
* DATA,I W * 1920 * 1941
31.0 28.2 32.2 37.0 37.0 38.6 40.7 41.5 42.9 45.3 42.1 39.3 34.3 34.1 36.6
39.3 44.2 47.7 45.9 49.4 53.0 61.8
* DATA,I Y * 1920 * 1941 +
<C+I+G-T>
* DATA,I E * 1920 * 1941 +
<Y+T-WG>
*
* TIME 1921 1941
* EQU,I C
C=F (P, P! 1!, W)
* EQU,I I

```

```
I=F (P, P ! 1 ! , K ! 1 ! )
* EQU,I WP
WP=F (E, E ! 1 ! , TIME)
* EQU,I Y
Y=C+I+G-T
* EQU,I P
P=Y-W
* EQU,I W
W=WP+WG
* EQU,I E
E=Y+T-WG
* EQU,I K
K=K ! 1 ! +I
*
* FILE,I KLEIN10
* COPY,D KLEIN., KLEIN10.
* FILE,B KLEIN
* FILE,D KLEIN10
* FILE,S KLEIN
* FILE,E KLEIN
* SYS,IS KLEIN
C
I
WP
Y
P
W
E
K
*
NO
YES
YES
* REP,ALL KLEIN
* END
```



## 第四章 世界経済モデルの構造とその解法

前章までの説明より、計量経済分析はデータ・マネージメントと推定とモデルの解法の三つの要素から構成されていることが明らかになったが、より大規模な構成のシステムの例として世界経済モデルを考えることにする。

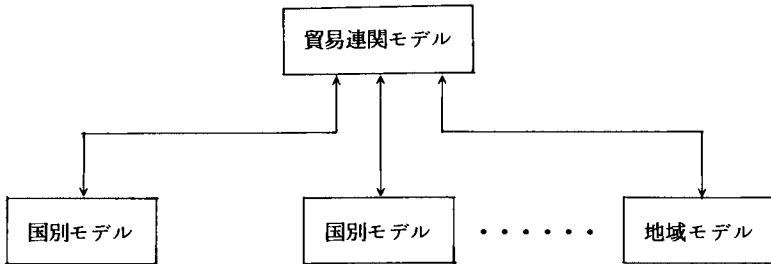
### 4.1 世界経済モデル

世界経済モデルの基礎的な考え方はそのまま部門別モデル（例えば財政、金融、生産等の部門に区分されたモデル）や地域間経済モデル（inter-regional or multi-regional econometric model）に適用することができる。一々断わっているとかえって混乱するので、本章では部門別モデルや地域間経済モデルには言及しない。以下、「国」、「輸出入」等はそれぞれ「地域」、「移出入」等と読みかえてもよいものとしておく。

世界経済モデルは通常、(1). 個々の国別ブロックと、(2). 国別モデル相互間の商品貿易を連結して、世界貿易および輸出入価格を整合的に決定する貿易関連ブロック、および(3). (1)(2)モデル間で必要な変数の授受および変換を行なう連結ブロックから構成される。これに、複雑なモデルになると、(4). 国別モデル間における経済変数の直接的関連を処理する直接関連ブロックを含むものを考えることができる。

ここでは最初、直接関連ブロックを含まないモデルについて考え、次にそれを含む場合について考えることにする。第4-1図に、世界経済モデルの構造を図式化して示しておく。

第4-1図 世界モデルの構造



#### 4.1.1 各国別モデル

各国についてのモデルは、個々の国の標準的な計量経済モデルである。本章は計量経済モデルについての高度の専門知識をもつ読者を想定しているので、国別の計量経済モデルについては説明しない。（世界経済モデルのための各国モデルについては、Ball RJ (ed) [1973] 等参照）

各国モデルを含めて、個々のブロックは、それぞれ独立にモデルの整備を行なうことが可能である。これら個々のブロックごとのモデルは、それぞれ独立に、十分な整備が行われ、シミュレーション結果が極めて良好なフィットを示していることが当然の前提になる。

#### 4.1.2 簡単なモデル

これらの国別モデルは貿易関連モデルによって相互に関連付けられる。そのもっとも重要な役割は、名目・実質のいずれかの面で、世界全体としての輸出と輸入の総計が等しいという整合性を保ちながら、世界貿易の水準を決定することにある。

もちろん現実には、評価方法（FOB, CIF等）の違いや、計上時期の相違などの理由で、世界輸出額と世界輸入額との間に統計上の不突合がある。整合性を考慮しながら貿易関連モデルを構成する方法としては、既に数多くの試みがなされている。この問題については、Waelbroeck [1973], Klein [1976]

等に詳しい。

いま、すべての変数を共通の通貨単位 (US\$) で表わすこととし、

$X_{ij}$  = 第  $i$  国から第  $j$  国への実質輸出

$XV_{ij}$  = 第  $i$  国から第  $j$  国への名目輸出

$X_i$  = 第  $i$  国の実質総輸出

$XV_i$  = 第  $i$  国の名目総輸出

$M_j$  = 第  $j$  国の実質総輸入

$MV_j$  = 第  $j$  国の名目総輸入

$TW$  = 世界の実質総輸出 (入)

$TWV$  = 世界の名目総輸出 (入)

とすれば、実質および名目での貿易関係は

$X_{11}$	$X_{12}$	$X_{13}$	.....	$X_{1n}$	$X_1$
$X_{21}$	$X_{22}$	$X_{23}$	.....	$X_{2n}$	$X_2$
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
$X_{n1}$	$X_{n2}$	$X_{n3}$	.....	$X_{nn}$	$X_n$
$M_1$	$M_2$	$M_3$	.....	$M_n$	$TW$
$XV_{11}$	$XV_{12}$	$XV_{13}$	.....	$XV_{1n}$	$XV_1$
$XV_{21}$	$XV_{22}$	$XV_{23}$	.....	$XV_{2n}$	$XV_2$
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
$XV_{n1}$	$XV_{n2}$	$XV_{n3}$	.....	$XV_{nn}$	$XV_n$
$MV_1$	$MV_2$	$MV_3$	.....	$MV_n$	$TWV$

で示される。ここで

$$X_i = \sum_j X_{ij}, \quad XVi = \sum_j XV_{ij}$$

$$M_j = \sum_i X_{ij}, \quad MV_j = \sum_i XV_{ij}$$

$$TW = \sum_i X_i = \sum_j M_j$$

$$TWV = \sum_i XVi = \sum_j MV_j$$

である。また

$PXi$  = 第  $i$  国の輸出価格

$PM_j$  = 第  $j$  国の輸入価格

$PTW$  = 世界輸出（入）価格

$a_{ij}$  = 第  $j$  国輸入市場における第  $i$  国の実質輸出シェア

$v_{ij}$  = 第  $j$  国輸入市場における第  $i$  国の名目輸出シェア

とすれば、

$$a_{ij} = X_{ij} / M_j \quad v_{ij} = XV_{ij} / MV_j$$

$$XV_{ij} = PX_i \cdot X_{ij} \quad XVi = PX_i \cdot X_i$$

$$MV_j = PM_j \cdot M_j$$

$$TWV = PTW \cdot TW$$

である。これらの式から

$$X_i = \sum_j a_{ij} \cdot M_j$$

$$PM_j = \sum_i (PX_i \cdot X_{ij}) / M_j = \sum_i a_{ij} \cdot PX_i$$

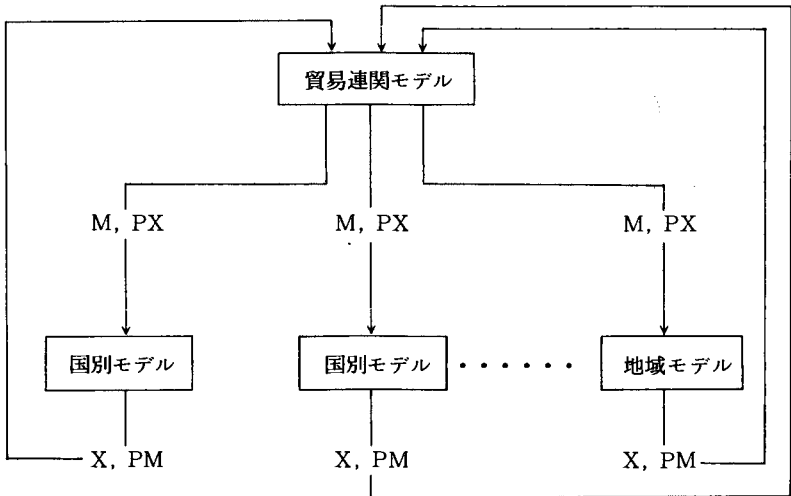
$$PTW = \sum_i (X_i / TW) PX_i$$

が導かれる。したがって、もし何らかの方法によつて、 $M_j$ ,  $PX_i$ , および  $a_{ij}$  ( $i, j = 1, 2, \dots, n$ ) が決定されれば、上式から、 $X_{ij}$ ,  $X_i$ ,  $PM_j$ ,  $TW$  および  $TWV$  が決定される。あるいは、 $PX_i$ ,  $MV_j$ , および  $v_{ij}$  が与えられれば、 $X_{ij}$ ,  $X_i$ ,  $MV_j$  等が決定される。言うまでもなく、このブロックの解法には、標準的計量経済モデルの解法を適用することが出来る。

標準的な計量経済モデルは、一般に輸入関数および輸出価格関数を持っているから、 $M_j$  (または  $MV_j$ ) と  $PX_i$  の値は、国別モデルを解くことによって与えられる。その値を貿易連関モデルのデータへ転送すればよい。したがって、問題は、 $a_{ij}$  (または  $v_{ij}$ )、つまり貿易シェア行列をどのように推定するかということになる。

この推定方法についても、既に多くの方法が試みられている (天野, 栗原, L. Samuelson [1980] 参照) が、本章では推定の問題には立ち入らない。これを図式化すると第 4-2 図のようになる。

第 4-2 図 貿易連関モデルを含む世界モデル





#### 4.1.3 直接連関ブロックを含む複雑なモデル

総合的な国際連関モデルを経由しないその他の国際的連関は、すべて各個別モデル間の直接的連関として処理される。例えば、各国モデルに現われる外国の諸変数（GNP、利子率、純資産残高等）は、該当する他国モデルの変数がそのまま用いられる。この場合は、変数の参照が貿易連関ブロックを含めて複数のブロックについて行なわれるので、やや複雑になる。

この場合、他国の変数はその国のモデルにとっては、必ず外生変数として取り扱われる。しかし、それらの外部変数の値を前提として求まったそのモデルの内生変数の値が、他国のモデルによって参照され、外生として参照した外部変数の値に影響するので、あまり多くの外部変数が互いに参照されている時は、すべてのモデルを一つにまとめて編集し、単一モデルとして解くのと、計算量は変わらないか、むしろ増大することになる。利点は国別の収束値の変動経過が得られるので、もし、モデルが発散したときは、原因がつかみやすいことである。

国別に同じ性質の変数（例えばGNP）には、同じ名称がつけられて参照に便利な様に整理してあることは、通常個々のモデルについては詳細な知識を持ち得ない LINK 部分の担当者にとっては重要な前処理である。しかし、他国モデルの変数を互いに参照するので、その変数がどの国に属するものであるかを容易に識別する何らかの措置が為されていないと、間違いと混乱の原因となる。これについては、自動的な識別処理がシステム側で配慮されていることが望ましい。

これらの変数名に逐一、ブロック別に異なる識別子（JAP.GNP, US.GNP, ITALY.GNP, etc.,）をつけて全変数を相互に識別可能な様に整理しておくものと規約するか、または、各国で同一の変数名を用いていても、ブロック自体に識別名がついていて、他のブロックに属する変数名を参照する時だけそれを識別子として変数名につけることとするかの二つの方法を考える

ことができる。当然、後者の方が、ユーザの負担が遥かに少なく使いやすいシステムになるが、システムの構造としては複雑になる。

しかし、たとえシステムがかなり複雑になるとしても、これについては、自動的な識別処理がシステム側で配慮されていることが望ましい。

## 4.2 世界経済モデルの解法

世界経済モデル分析システムは、前節でのべた諸機能を持った仮想電子計算機とその言語を作成して結合すればよいことになる。このためのシステムとしては、いろいろな構成方法があるが、大別すれば、

- (1) 全体を単一モデルとして解く仮想電子計算機
- (2) 各国モデルを解く各々の仮想電子計算機とそれらの仮想電子計算機を順次呼び出しながら貿易連関モデルを解く仮想電子計算機との複合仮想電子計算機
- (3) その他

の三つに分類できる。

### 4.2.1 第一の方法

この方法はシステムに含まれる総ての各国モデルおよび貿易連関モデルを並べて記述し、一つのモデルに編集してしまう方法である。世界経済モデルを構成する総ての式が一つの連立方程式に含まれてしまっていることになるので、全ての式が同時に解けるだけの大きさの電子計算機が利用できれば、比較的簡単に作成可能である。電子計算機の知識を充分に持たない計量経済学者が、世界経済モデルの分析が出来るプログラムを自力で書き上げるにも、あまり高級な技法を使わなくてすむ。

これは、単に各国モデルをつなぎ併せてゆけばよく、電子計算機の性能や利用条件に制約がなければ、案外、実際的な手法だと言える。解法の算法（アルゴリズム）は一般の（一国の計量経済モデルの）場合と全く同じである。

ただし、この方法では、上述したように、世界経済モデルに含まれる膨大な数の全変数名が、互いに識別可能ようになっていなければならないし、モデルの管理や改良・訂正、さらに、効率的にシミュレーションを行なうこと自体が、モデルの規模の拡張につれて累積的に困難になる。また例えば、貿易シェア行列の推定方法を複数試みて、結果を対比してみたいと言う様な場合、ケース毎にモデルの膨大なプログラムから成る仮想電子計算機の構成を全体から書き直すことも必要になる。

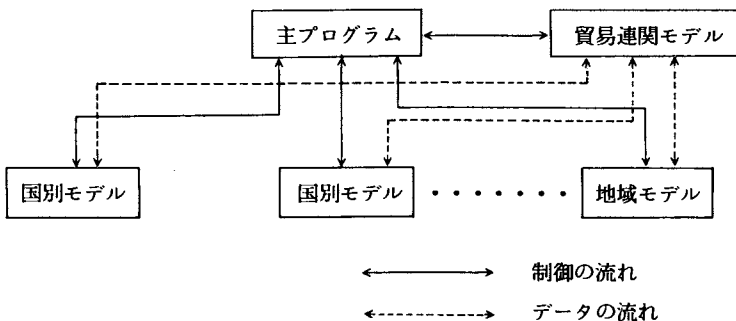
#### 4.2.2 第二の方法

個々のブロックを（独立の）仮想電子計算機として構成しておき、別に一つの制御用の仮想電子計算機を置いて、それで全体をシステムとして構成する方法である。この各々の仮想電子計算機がFORTRAN 言語で記述されなければならないシステムの例として、プロジェクト・リンクをはじめ、多くの採用例がある。第4-3図にこの方法の構造の概要を示す。

第一の方法との算法上の主な相違点は、繰り返し計算によって連立方程式を解いてゆく場合の解法の打ち切り条件にある。

第一の方法では総ての各国モデルに含まれている全（内生）変数が、原則として等しいウェイトで評価され、それらの変化が同時にある既定範囲内に落ち

第4-3図 一般的な（Project LINK等）プログラム構成



着く様になった段階で、解が収束したと判定する。しかし、一部が収束しないだけでも、その影響は全体に波及することがあり、しかもそうなれば、どこで不都合が発生しているのかを発見することは、しばしば困難である。

これに対して、第二の方法では、解法の打ち切りは、各国モデル毎に条件が設定され、他の国のモデルから与えられる外部変数に対応して、各国モデル内の内生変数の変化がある既定範囲内に収まる様になれば、解法は一応打ち切られる。この方法の場合、システム全体での解法の打ち切り条件は、貿易関連ブロックの内生変数の変化が、ある既定範囲を超えなくなった時である。

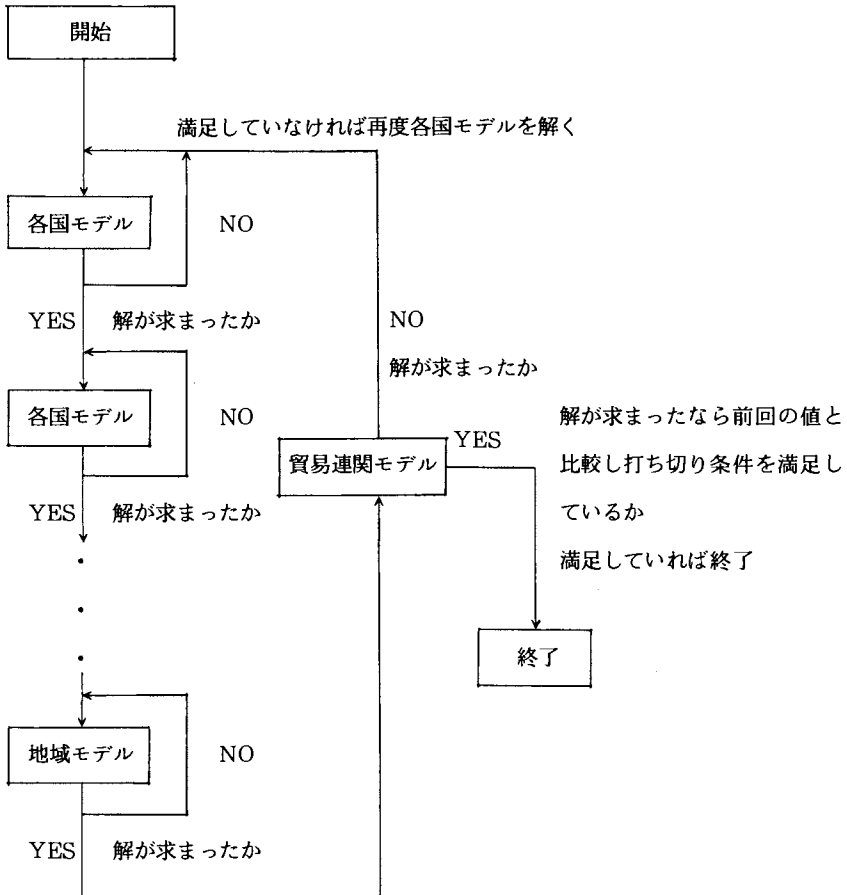
詳しく説明すると、先ず、ある初期条件（初期値）を与えて各国モデル用の仮想電子計算機で各国モデルを個別に解く。その結果として得られた各国の実質総輸入または名目総輸入（ $M_j$ または $MV_j$ ）と輸出価格（ $PX_i$ ）の値を、すべて貿易関連ブロック用の仮想電子計算機に転送し、これらの値を外生変数として使用して貿易関連モデルを解く。

かくして貿易関連モデル内で各国間の実質輸出（ $X_{ij}$ ）、各国の実質総輸出（ $X_i$ ）、各国の輸入価格（ $PM_j$ ）、世界の実質総輸出（入）（ $TW$ ）および世界の名目総輸出（入）（ $TWV$ ）が新しく決定される。この結果は各国別モデル用の仮想電子計算機へ再転送され、先の初期条件（初期値）に代えてこの値を使用して、再度各国モデルを解き、この結果の値をさらに貿易関連モデルに戻してこのモデルを解く。

この繰り返し計算を何度も繰り返し、各国間の実質輸出（ $X_{ij}$ ）、各国の実質総輸出（ $X_i$ ）、各国の輸入価格（ $PM_j$ ）、世界の実質総輸出（入）（ $TW$ ）および世界の名目総輸出（入）（ $TWV$ ）が前回の値と比較して変化が一定範囲内に収束するまで繰り返す。これを、図式化すると第4-4図ようになる。

この方法では、各国用の仮想電子計算機を作成するのは、第一の方法と同じであるが、これらの仮想電子計算機を制御するための仮想電子計算機を作成するのに、電子計算機プログラミングのかなり高級な技術を使わねばならない。

第4-4図 一般的な制御の流れ



電子計算機の知識のない計量経済学の専門家が自分でプログラムを組んで分析を行なうのには、相当難しい手法だといえる。

もっとも、誰にでも容易に利用出来るような、世界経済モデル分析専用のシステムを開発するのに、あまりたいした問題はない。貿易シェア行列については、推定方法ごとに別仮想電子計算機を用意することになる。

第一の方法に対してこの方法は、各国毎に独立に計算が行なわれるので、収

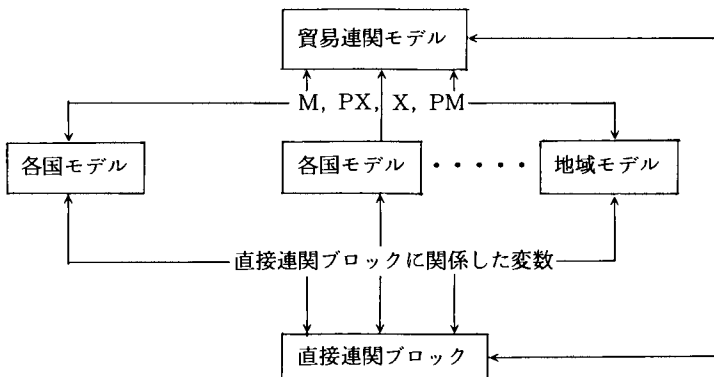
束が悪い場合、どの国のモデルに問題があるのかが容易に見発できる。フィットのよい国モデルの計算はすぐ打ち切られるから、計算時間も大幅に短縮される。管理も各国毎に行なうので比較的容易である。

前節で述べた直接関連ブロックを含む場合は、第一の方法によれば簡単に実現可能であるが、第二の方法による場合は、各国間でやりとりする変数をどう取り扱うかを考えなければならない（第4-5図参照）。さらに、モデルの維持・拡張の点から、各国モデルに現われる外国の諸変数に変化があるとシステムそのものの変更を必要とする場合も生じて来る。

4.2.3 第三の方法

以上第一、第二の方法には、一長一短があり、他にもいくつかの解法を考えることができる。京都大学東南アジア研究センターでは、上記とは違う第三の方法を採用することにした。各国モデルと貿易関連ブロックを、それぞれ仮想電子計算機とみなす所までは第二の方法と同一であるが各国モデル相互間、貿易関連ブロックと各国モデル間のデータの転送については、データ・マネジメントのための仮想電子計算機を配置する方法である。その詳細については、

第4-5図 直接関連ブロックを含む世界モデル



次章以降に詳述する。

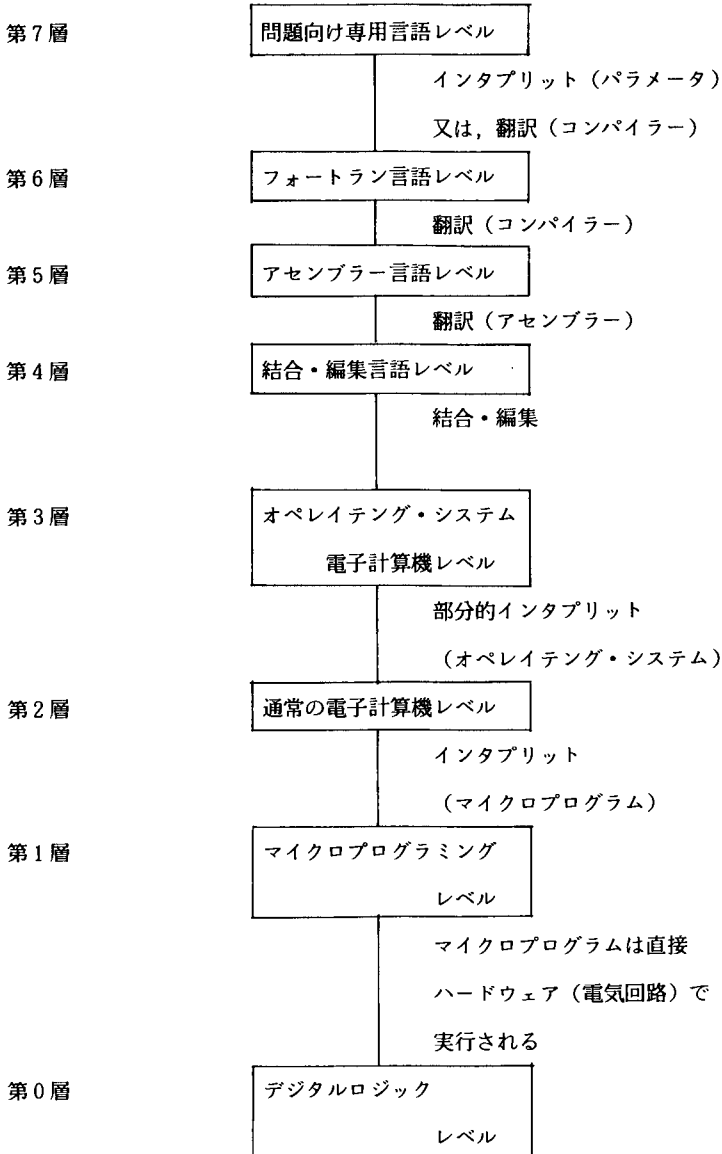
### 4.3 世界モデルの結合編集

第二、第三の方法を現存する実電子計算機で実現するためには、これらの並列に配置された仮想電子計算機を有機的に結合する必要がある。つまり、今現存する汎用電子計算機の上にこれらの複数の仮想電子計算機を構築した場合、これらの仮想電子計算機の実行の順序（および、必要ならば電子計算機資源の割り当て）等の制御を行なう必要がある。通常これらの機能のために汎用電子計算機では、結合・編集機能（LINKAGE EDITER）を備えている。

結合・編集機能を仮想電子計算機として捉えた場合、その機能は複数の仮想電子計算機を一つの仮想電子計算機に編集するための仮想電子計算機であるといえる。つまり今まで議論してきた仮想電子計算機の層数が上部に追加されるのではなく下部に追加された形になる（第4-6図参照）。

この仮想電子計算機は当然、各仮想電子計算機間での共通データの受渡し機能ももつことになる。それが、実際にデータを転送するのか、それとも一定の共有データ領域（COMMON data area）を設定して、各仮想電子計算機はそれへアクセスするようになるのかは、設計者にとっては、単に便宜上の問題だけである。

第4-6図 結合・編集機能を持った仮想電子計算機







## 第五章 世界経済モデルの解法システム

世界経済モデルといえど、もし各国モデルが独立していないで、全体が一つのモデルとして編集されていれば、第4-2図からも分かるように、仮想電子計算機としては、一般的な計量経済モデルの仮想電子計算機と同様とみなすことができる。さらに言えば、一般的な計量経済モデルで、所得決定部門、生産部門、金融部門、雇用部門等のサブモデルに区分して考えるのと本質的に大差ない。

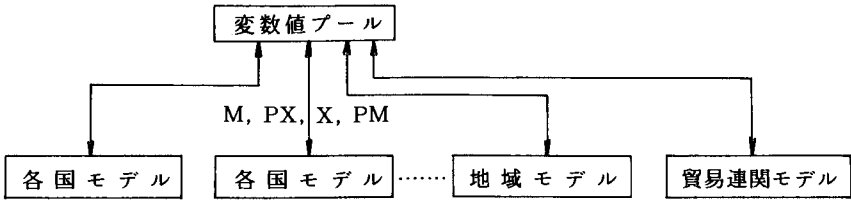
### 5.1. 単一モデル解法システムと世界モデル解法システム

このことに注目して、計量経済モデルの解法システムを、第5-1図のように変形することを考えてみる。

今、システム内では、同時には一つのモデル・ブロックだけを対照に分析する（一つの仮想電子計算機だけが実行される）と仮定する。ただしこの場合、全モデルに含まれるすべての変数は、システムによって一元的に管理されており、計算中のモデルが必要とする外部モデルに属する変数（そのモデルの外生変数）は、必要に応じてシステムが自動的に、他のモデルのデータ領域からそのモデル・ブロックのデータ領域に供給するように設計する。さらに、そのモデルの解が求まった時、そのモデルの解として得られた変数（そのモデルの内生変数）の値は自動的に更新されてシステムに記憶され、引き続いて行われる計算過程で、次に他のブロックからその変数が要求された時には、最新の値が自動的に供給される様にする。この様な機能を持った仮想電子計算機を考えることにする。

つまり、世界経済モデルの解法がスタートすると、世界経済モデルに含まれ

第5-1図 第三の方法による世界モデル



(注) 変数値プールは、システムが必要とする時に  
変数の値を各モデルに渡す働きをする。

ている全変数について、その名称(変数名)および値が、一時的なファイルに移され、他の個々のモデル・ブロックおよびそのデータとは独立のデータ・ブロックによって一元的に管理される。モデル・ブロックのデータ形式が、通常1変数につき1レコードの時系列データであるのに対して、この一時的ファイル内のデータは、1期についてその期の全変数の値をもって1レコードを形成する横断面的(cross-sectional)データである。

この一時ファイル内のデータは、どの仮想電子計算機からでも、システムに登録されている変数名をつかってデータを求めた場合は、その値を要求先の仮想電子計算機へ転送し、反対に、ある仮想電子計算機が、登録変数名をもつデータを計算・加工した場合は、その値をかならずこの一時ファイルへ記録しておく様にする。それが分析中を通じて変化しないか、または、先に解かれた仮想電子計算機の解として値が変わってしまっているかは、システムは問題としない。内容は常に最新のものに更新され、要求があれば、該当する変数の最新の内容が自動的に供給されるものとする。

以上のデータ管理機能の他に、さらに、

- (1) 個々のモデル解法の実行される順序を制御する機能
- (2) 個々のモデルの解の打ち切り条件を制御する機能

があれば、第二の方法と同様に世界経済モデルが解けることになる。もし、上記の機能を何らかの方法で実現出来たと仮定すると、貿易連関モデルを始め各国モデルを、各々単一仮想電子計算機と考へて、今までどおりに、個々に、記述すればよいことになる。

先に述べた各個別モデルごとの変数名の取り扱ひについては、次の様な処理を考へることが出来る。

各ブロックは、通常、独立にモデル作成が行なわれ、データファイルもブロック別に編集されている場合が多い。そこで、各ブロックごとに、モデルの specification を格納したファイルおよびそのデータ・ファイルに識別名をつける。利用者は、外部ブロックに属する変数名を参照する場合は、変数名の前に、この識別名を識別子として、ピリオド( . )をつけて付加する規約とする。たとえば、アメリカ・モデルの識別名をUSとして、他のモデルからアメリカ・モデルに属する変数、GNPを参照したいときは、変数名をUS.GNPと書くこととする。

各ブロックでのモデル作成を行うには、既存の計量経済分析システムについて、変数名の書き方についての処理部分を識別子を付けられるように改定するだけで、外部のモデルに属する変数を外生変数として取り扱うことにより、独立に作業することが出来る。

複数のモデルのリンクを行う場合は、[1] 一つのサブ・モデルの処理を始める場合、そのサブ・モデルの含む変数名を調べて外部モデル識別子のついたものがあれば、そのシステム・ファイル中の外部モデルのデータ・ファイルから、該当する変数のデータを、処理中のモデルのデータ・ファイルに転送し、そのサブ・モデルの解が得られた時はその解でシステム・ファイルのそのモデルの対応する諸内生変数を更新する仮想電子計算機機能を、既存の計量経済分析システムに追加する。さらに、[2] 各モデル・ブロックの解を連続して求めてゆくバッチ ( BATCH ) 処理が実行可能な様に、システム機能を付加する。

かくして、専用のモデル記述言語を持った優れた単一計量経済モデル分析シ

システムならば、すべて、若干の機能を追加するだけで世界経済モデル分析システムとして利用することができる。

## 5.2 世界経済モデル・システムの作成

われわれはこの点に注目して、第5-2図に示す様な世界経済モデルの処理システムを、STS (Stochastic Simulation System) をベースとして構築した。

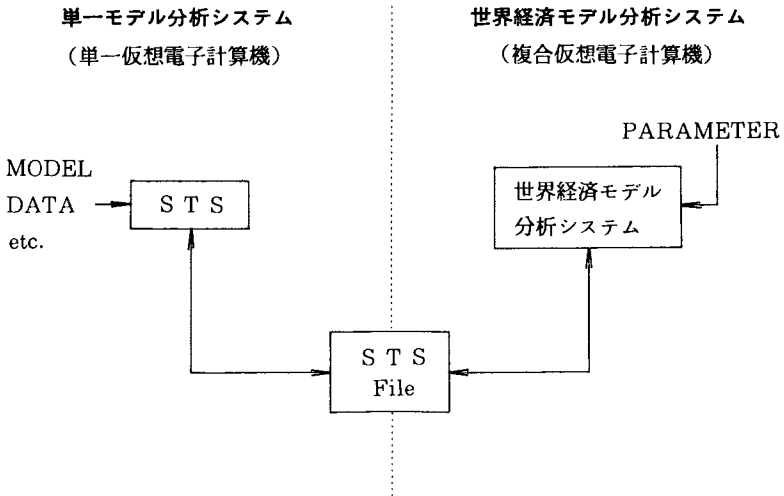
個別モデルの開発・作成に当たっては、単一計量経済モデルの分析システムであるSTSを、そのまま使用するものとした。データ・バンク機能もSTSのものをそのまま流用して、これに、データ統合管理機能、およびバッチ処理機能を実行するための仮想電子計算機を追加した。上記の諸機能の他にも、若干、必要な機能があるが、基本的な機能は、以上である(細部については、次節以降参照)。

われわれの方法では、変数値については、個々のモデルが変数を参照する場合に、その変数名が外部識別子を持っていれば、先に解かれた他のモデルの解の値を、自動的に検索して対応する。この機能に基づいて、モデルを解く場合の制御を次のように拡大して行なえる様にすることによって、自由度の高い対応をすることができる。

- (1) 各ブロックを解いてゆく順序は、実行時に仮想電子計算機に与えるパラメータにより、自由に制御することができる。
- (2) 各仮想電子計算機間の変数値の授受は貿易関連モデルの各国モデルとの間だけでなく、一般にどの仮想電子計算機間でもできる。
- (3) 各仮想電子計算機毎に解の打ち切り条件の制御を行うことができる。

以上の処理によって、直接関連ブロックを含む場合でも、単一の方程式システムに編集する場合と同様に、特別な考慮を必要としないで、モデルを構築する

第5-2図 世界経済モデル分析システム



ことが出来る。以上の諸機能により、モデルを解くに当たって仮想電子計算機制御の方式さえ適当に工夫すれば、現在世界各地で行われている世界経済モデルの分析手法については殆ど全て分析可能であると言える。

例えば、A国とB国の間で互いに相手国のGNP等を各々の国別モデルのなかで使用している場合、まず、ある初期条件（初期値）を与えてA国を解き、次にその値を使用してB国を解く。次にこの結果の値を使用して再度A国を解く。この繰り返しを何度も繰り返し、貿易関連モデルの場合と同様にGNP等の値が前回の値と比較して変化が一定範囲内に収束するまで繰り返す。更にC国がA又はB国のGNP等の値を使用している場合（三層間の場合）、A、B国の二層間の解が求めれば次にC国を解き、C国の解が求めれば、この値を使用して再度A国およびB国を解く。そして上述した手順で、もう一度A、B国の二層間の解を求める。そしてC国に戻る。以上の繰り返しによって直接関連ブロックを含む場合でも、直接関連ブロックのモデルを作成することなく解く事が可能

である。この様子を第5-3図に示す。

### 5.3 世界計量経済モデル分析システムの開発手順

より一般的に、世界計量経済モデル分析システムの開発方法について説明しておく。すでに述べた様に、専用のモデル記述言語をもったシステムなら、ほとんどの計量経済モデル・システムは、以下の改定によって世界計量経済モデル・システムに転換することが出来る。

#### [1] 個別モデル用ファイル名の付け方の規約変更

変更対象となるシステムは、通常、単一モデルの分析時にデータ・ファイルや、モデル記述ファイル等の各種の中間ファイルを生成する筈である。同じシステム・デヴァイスを使用して複数のモデルを処理できるように、一つのモデルで使われる全ファイルを、他のモデル用のファイルから識別できるように、そのモデル固有の識別名称を、与えることが出来る様になっている場合は、問題はない。もし、各ファイルに、システム固有の特定ファイル名が付けられているなら、それを、モデル毎に、識別可能な共通ファイル名を付けることができるように変更しなければならない。

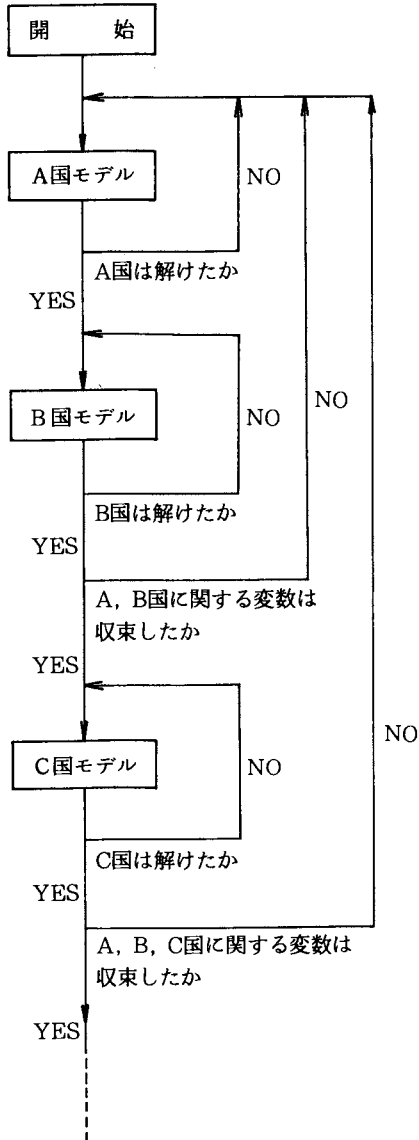
ファイル・名は、`cccccc.ccc`の型で与えるようになっており、ピリオドで区分して、2つの部分に分けて命名できるようになっている。この前の部分を共通ファイル名すなわちモデル名（例えばJAPAN）に、後の部分をファイルの種類の識別（例えば、データ・ファイルの場合、DAT）に用いることとする。これによって、例えば、日本の単一モデルについての各種ファイルは、

データ・ファイル            JAPAN. DAT

モデル・ファイル            JAPAN. MDL

というような形の名称が付けられ、他のモデル用の各ファイルと区別しうるようになる。

第5-3図 直接関連ブロックを含む場合の制御の流れ





## [2] 変数名の付け方の規約変更

まずシステム全体について、変数名に関する規約を、識別子を付けられる様に変更する。システムの殆ど全体に及んで変数名をチェックする処理を含んでいるのが普通なので、これがおそらく一番面倒な変更であろう。識別子の区別記号はピリオド { . } でなければならないと言う訳ではない。一部のシステムでは、ピリオドを特別な機能に使っているため変数名に使える様にするのが困難な場合があるが、こんな時は他の記号 {たとえば, @, !} をそれに充てればよい。

この変更によって、たとえば、

US.GNP, JAP.C, UK.WP, etc.,

を変数名として使用することが出来るようにする。

## [3] システム統合データ・ファイル編集ルーティンの作成

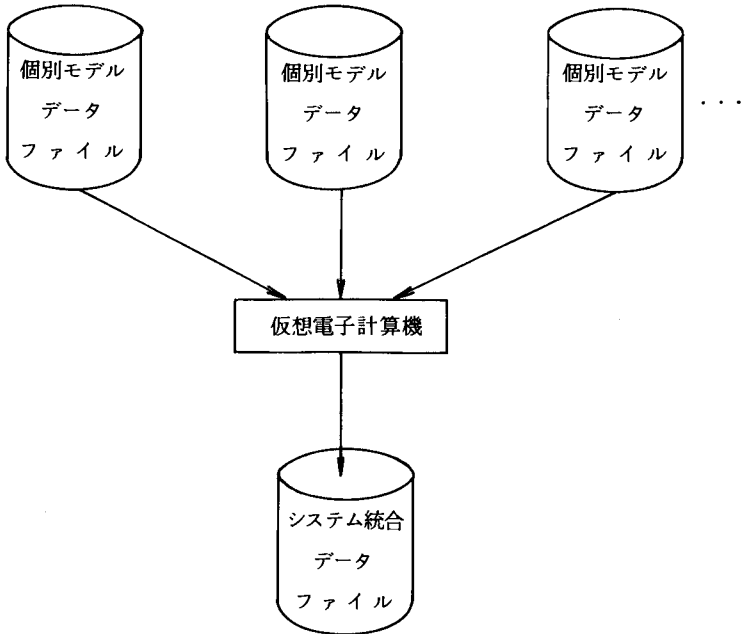
システムに新しい仮想電子計算機を付加する。これの持つべき機能は、第5-4図に示すように、個別モデル(国別モデル)ごとのデータ・ファイル(変数名ファイルを含む)を一つに編集して、システム統合データ・ファイルを作成することである。

この統合データ・ファイルの変数名表には、個別モデルの変数名で識別子の付いていないものには、そのファイルのモデル名を識別子として、区分記号を付けて登録し、識別子のついているものについては、すべての個別モデルについて、当該他モデル・データと照合して、対応するものが同じであれば、消去し、同じでなければエラー・メッセージをだすようにプログラムする。

なお、識別子つき変数名については、後述するシミュレーション・プログラムの変更仕様との関係で、2つのモデルのデータ・エリアのそれぞれに二重登録しておくことも考えられる。

データは、計量経済モデル分析システムでは、普通、時系列的に1変数分のデータが1レコードに入っていると思われるが、これを全モデルの変数が、1期につき1(論理)レコードに収容されるように変更する。

第5-4図 システム統合データ・ファイル編集仮想電子計算機

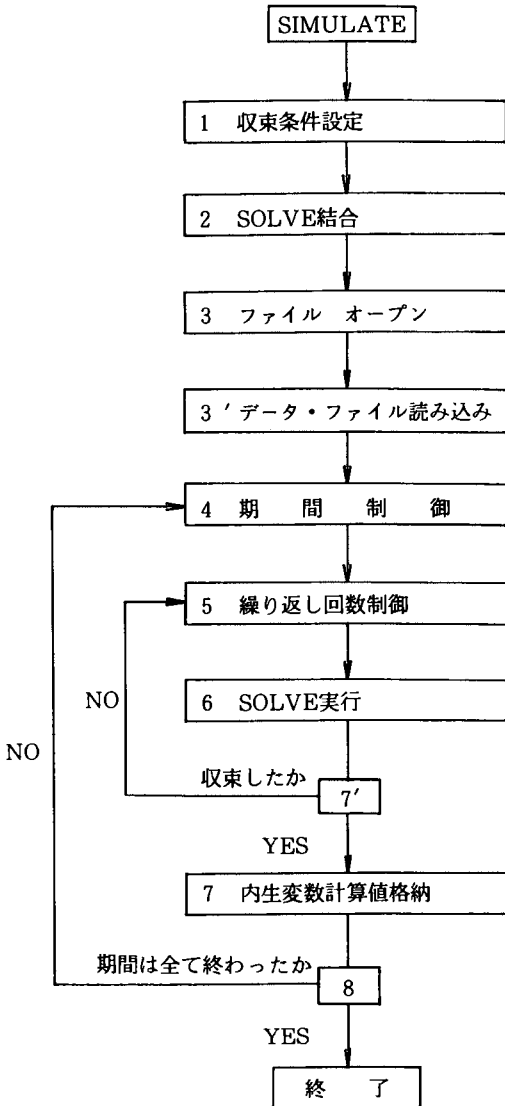


このプログラムは、システムの他の部分とは独立に作成しうる筈である。また、強いてシステムに組み込まなくても、使用頻度は少ないから、必要に応じて独立に動かしてもよいし、また一度作成すればかなり長期に使用することになるので、効率的なプログラムを組むのに苦労する必要もないだろう。ただ、シミュレーションを実行するたびに、もとの内容は変更されてしまうので、必ず backup ファイルをつくり、シミュレーションを行う場合はこの backup ファイルをコピーしたものを使用するように、プログラムしておかねばならない。

#### [4] シミュレーション・モジュールの変更

計量経済モデル分析システムのシミュレーションを実行する部分には、大幅な変更を加えねばならない。この部分は通常第5-5図に示すような流れになっ

第5-5図 一般のシミュレーション・プログラム



ている。

第1ブロックでは、シミュレーション期間、収束条件等について、利用者に設定を求める。

第2ブロックでは、前の段階で処理された、モデルの実行可能な形式のプログラム (SUBROUTINE SOLVE) とプログラムの本体部分とが結合される。これには、システムが記述されている言語の性格に応じて色々な様式がある。

第3ブロックでは、対象となるモデルについての各種のファイルが open され、パラメータ・テーブル、変数名テーブル (および、一部のシステムではデータも) 等が読み込まれる。

第4ブロックは、期間の変化を制御する部分である。この部分ではまた、データが最初に一括入力される形式のシステムでないときは、当該のデータを読み込む。

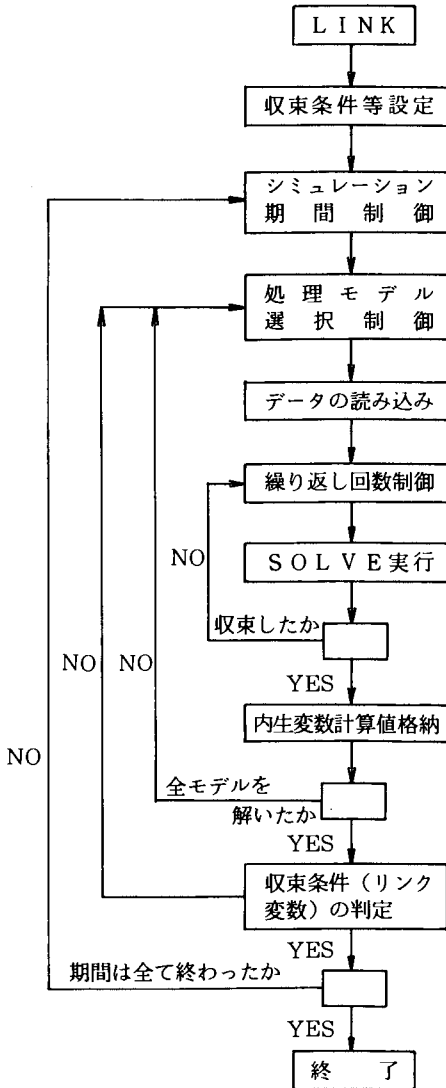
第5ブロックは、収束計算の回数を制御する。この部分は第2ブロックで設定された収束条件に基づき、第5ブロックの収束結果判定プログラムと協同して、繰り返し回数を決定してゆく。

第6ブロックは、SUBROUTINE SOLVE (この中には、第三章のペンシルバニア大学のシステムで説明した CONST, SOLVE, POSTR が含まれる) である。連立方程式の繰り返し解法が1回あるいは数回実行される。

第7ブロックは、収束結果判定ブロックで、設定された条件によって、計算結果を判定する。収束未了の場合は第5ブロックにもどって、計算を繰り返す。収束が終了した場合、または、設定された最大繰り返し回数をすぎた場合には、次に移る。データが最初に一括転送される形成でない場合は、計算された内生変数の値をファイルに格納する。

第8ブロックは、第4ブロックと共に、解法実施期間を制御する。指定された全期間について計算が完了すれば、シミュレーション部分の演算は終了し次の過程、たとえば、シミュレーション結果の表示に移る。

第5-6図 リンクの場合のシミュレーション・プログラム



以上のプログラム対して、第5-6図の様な変更を加える。つまり処理モデル選択制御、リンク変数に関する収束条件の判定を追加し、一部の処理順序を変更すれば、通常モデル分析システムでリンク・モデルを解くことが可能である。



## 第六章 東南アジア研究センターのシステム

京都大学東南アジア研究センターの世界経済モデル（このモデル等については、Ichimura S. Ezaki M. [1985] 参照）の解法プログラム・システムは、ベースとなる単一モデル解法システムに STS (Stochastic Simulation System) を利用して開発した。

### 6.1 各国別モデルの準備

各国別のモデルは、STSによって推定され、特化されて、変数名表および変数と共に、ファイルに格納される。各国別に厳重なfinal test を実施しておかねばならないことは言うまでもない。Link を行なう部分（今後、便宜的にTRADE model と呼ぶ）も、一国のモデルと同等に取り扱い、必要な、変数名表、方程式を、予めテストしてファイルに入れておく。

各モデルには、8字以内の識別名 [identifier] を定め、互いに区別しうる様にしておく。識別名の例を下に示す。

JAPAN, KOREA, USA, TAIWAN, PHILIP, THAI, etc.

TRADE model と各国モデルとの間で転送しなければならない変数については、変数名は双方のモデルを通じて同じでなければならない。反対に、一つのモデル内でのみ用いられ他に転送されない変数は、その国別モデル内だけで互いに独立な名前が付けられていけばよいが、国別モデル間でデータ転送を行なう場合は、同様の考慮が必要になる。

このシステムでは、プログラム・リンク用コンバーターにより、このシステムによって作成されたモデルを、実行可能な形に処理する。この際、各国モデルと TRADE model との間のデータ受け渡しの整合性がチェックされる。



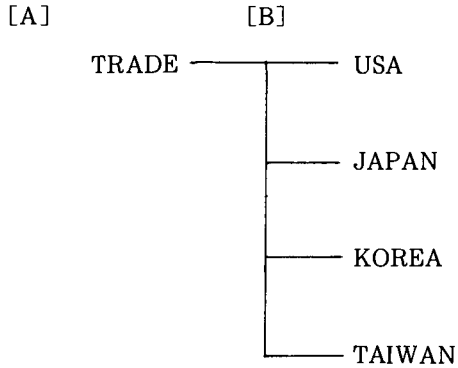
つまり、システムは統一された変数名により、各変数がどのモデルから参照されているかを認識する。このデータ転送に必要な情報も、プログラム化されて、実行ルーティンに渡される。

## 6.2 解の順番の指定

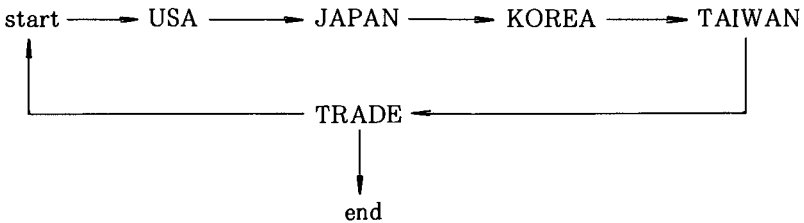
この様にして準備された、世界および各国モデルについて、それをどんな順序で解いてゆくかをシステムに指示しなければならない。本システムでは、指定する方法として、プログラミングで使用されている 'OVERLAY' の表記方式を準用した。この方式で全体モデルの構造を記述すると、次の様になる。

Model Name	Segment Name
TRADE	A
USA	B
JAPAN	B
KOREA	B
TAIWAN	B

この linked model は、USA, JAPAN, KOREA, TAIWAN の4つのモデルを結合しようとしている。その他の世界についての情報は、TRADE に一括されており、4つのモデル相互間の依存関係も、総て、TRADE に記述される。segment には、各モデルを 'branch' と考えた場合の 'root name' が書かれる。上の例について言えば、USA, JAPAN, KOREA, TAIWAN は、同じ root [B] から出た枝であることを示す。この場合 root は、すぐ上にあるモデル TRADE である。TRADE は segment 名としてAを持ち、他のモデルとは別のグループに属するものと見なされる。そして、上位のモデルが存在しないので、root は存在しない。



モデルの解かれる順序は、まず、branch にあるモデルについて、同一の segment に属するモデルを前から順に解いてゆき、終われば、root にある上位のモデルを解く。これを各期毎に解を満足するまで繰り返す。

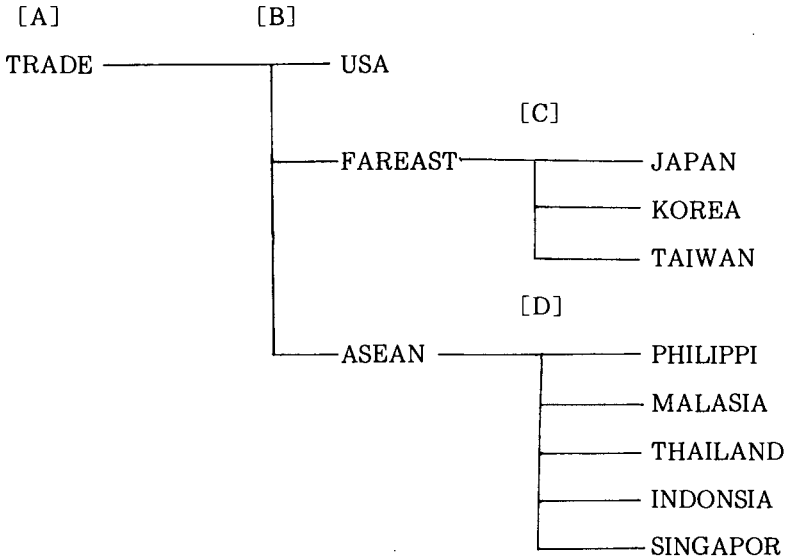


次に示す例は、多重イテレーションの場合である。この例では各モデルは、複数の SUB TRADE MODEL に GROUPING され、その中でより強い相互関係を持っているものと見なされる。

Model Name	Segment Name
TRADE	A
USA	B
FAREAST	B
JAPAN	C
KOREA	C
TAIWAN	C

ASEAN	B
PHLLIPPI	C
MALASIA	D
THAILAND	D
INDONSIA	D
SINGAPOR	D

この場合 MODEL 全体の構造は次の様になる。



システムは、まず、JAPAN - KOREA - TAIWAN と FAREAST とを解き、次いで PHILIPPI - MALASIA - THAILAND - INDONSIA - SINGAPOR と ASEAN とを解く。これらについて解が得られれば、USA - FAREAST - ASEAN を解く。これらについて解が得られれば、USA - FAREAST - ASEAN を TRADE と連結して解く。その結果は再び各サブ・グループに返され、全体に満足すべき解が得られるまで、iterate される。各分岐を辿ってその末端からモデルの解を求めてゆき、同じレベルの

segment が終わると上位モデルとの間でデータ転送を行なってこれを解く。結果は又、末端のモデルに転送されて、iteration が繰り返される。一つの分岐が完了すると、次の分岐に移る。

この様にして、model name に segment name を付けて、分岐の状態を表示すれば、システムはそれを解析して、方程式を解いてゆく順序および連結して解く範囲を自動的に判別して計算を遂行してゆく。分岐状態の表示は、容易に記述出来るから、組み合わせによって自由に LINK MODEL 全体の解き方を決定することができる。もっとも、分岐の段階が多くなってくれるにつれ、一つの期について、各モデルを何回も何回も解かないと最終的な解が得られないことになる。従って実際的には、3, 4 段階が限度と言うことになるが、実用的にはそれで十分有用である。

### 6.3 データ転送用テーブル

各モデルの持っている各種情報を解析して、各モデル毎に変数の受け渡し表が作成される。本システムの場合は、STS の持っているテーブルを拡張して、次の様な表を各モデル毎に作る。Variable Transfer Table は、link 用の変数を登録しておく表であるが、これを作成するには、各 branch の末端から最上位の root までチェックしなければならない。上位の root model で長期にわたる time lag がある変数が link 用の変数として使われている場合は、対応表だけでなく、モデルを解く期間等のパラメータも、この lag を満足する様に変更せねばならない。

変数の転送には、各モデルの内生変数を上位モデルの外生変数として送る場合と、下位モデルの外生変数として送る場合とがある。いずれの場合にも、長期の time lag がある場合、そして特に多段階に分岐している場合には、チェックの為のプログラムはかなり複雑なものになる。まして、同位のモデル間での転送を認めるとなると、始末に終えなくなる。この為、本システムでは

現在までのところ、同位のモデル間での変数転送を認めないもの規約している。さらに、既に述べた様に、多段階になればなる程、iteration の nest が累積的に増大して、且つ、それだけ解が得られなくなる可能性も増大する。

第6-1図 データ転送テーブル

PARAMETER TABLE Model Name, Number of Variables etc.
Endogeneous Variables Name Table
Exogeneous Variables Name Table
Variables Transfer Table to & from Other Model (s)
Model Specification

## 第七章 分散型仮想電子計算機

前章の説明からもわかる様に、世界計量経済分析システムは、単一仮想電子計算機と複数の仮想電子計算機の複合体と見なすことができる。図式化すると第7-1図の様になる。京都大学東南アジア研究センターのシステムを例にとって説明すると、各国モデルと貿易関連モデル（ブロック）を作成するのにSTSという仮想電子計算機を使用し、さらにこの仮想電子計算機の結果を利用して第五章で解説した複数仮想電子計算機で世界計量経済モデルを解くことになる。つまり二つの仮想電子計算機を、ファイルという手法で有機的に結合したシステムとなる。

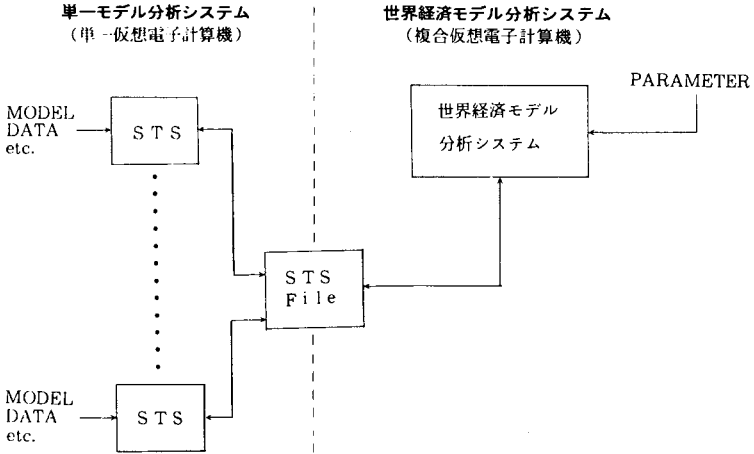
このファイルという手法は、機能的に見れば、ただ単にデータの転送を行なっているにすぎない。このことに注目すると、これら複数の複合体仮想電子計算機を、さらにこれら全てを包含した仮想電子計算機として捉えることが可能となる。つまり大型システムを、複数の並列に配置した仮想電子計算機の集合体として捉えることが可能である。

### 7.1 複合体仮想電子計算機

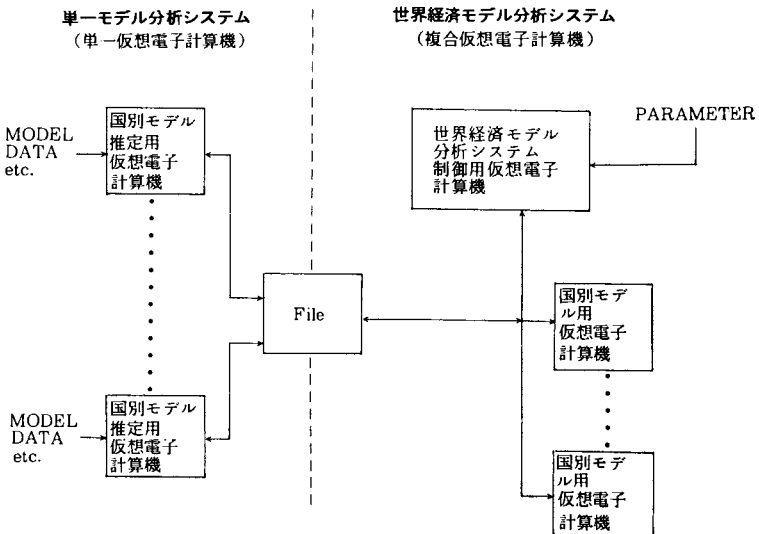
第四、五、六章で世界計量経済モデル分析のためのシステムの構成方法について考察したが、さらにこれら全体を仮想電子計算機として図式化すると第7-2図のようになる。この複合体仮想電子計算機は、各々の仮想電子計算機を有機的に結合するためにファイルという外部媒体を利用しているが、データの流れにだけ注目すれば、その機能を他の電子計算機（仮想電子計算機）で代替することが可能である。この様子を図式化すると第7-3図の様になる。

さらにこの仮想電子計算機全体を、データの流れにだけに注目して模式化す

第7-1図 世界計量経済分析システムの例



第7-2図 世界計量経済分析システムのための仮想電子計算機



ると第7-4図の様になる。つまり中心にワーキング・データの一時保管・転送だけを行なう仮想電子計算機があり、その周辺に各国モデルや貿易連関モデル（ブロック）を解くためのそれぞれの仮想電子計算機が配置されている形になる。

この考え方をさらに一般化すると、ワーキング・データの一時保管・転送を行なう仮想電子計算機の回りに単一機能を持った仮想電子計算機が、その機能の数だけ配置されている複合体仮想電子計算機を考えることができる。その各々の仮想電子計算機は言語システムであることもあるし、結合・編集機能のシステムであることもあるし、推定機能だけのシステムであることもある。場合によっては、各々の国モデルの解法システムであることもある。

計量経済分析システムだけに留まらず、一般のシステムについても同様な模式化が可能である。

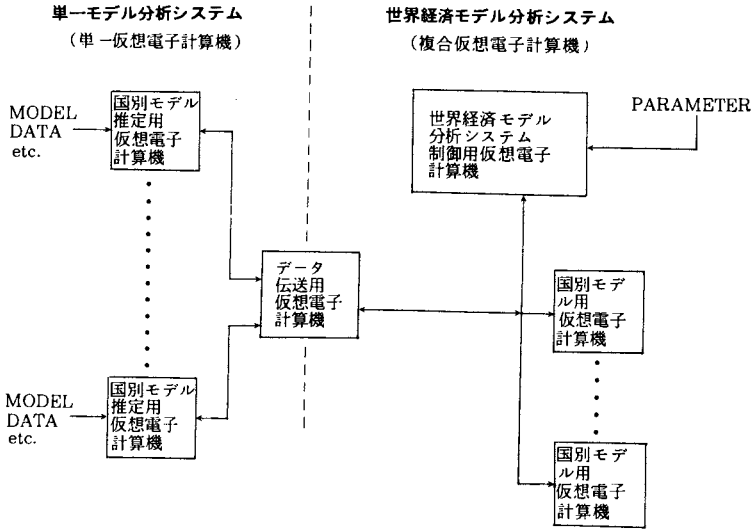
## 7.2 データベース・マネージメント・システム

京都大学東南アジア研究センターの世界計量経済モデル分析システムの場合、中央に位置する仮想電子計算機は、ワーキング・データの一時保管・転送だけを行なうものとしたが、さらに拡張して、それ自身にデータの蓄積・加工・検索・表示その他の一般的なデータ管理機能を持たせたシステムを考えることができる。つまり、大規模なシステムを設計する場合、データ・ベース・マネージメント・システム（DBMS）を、まずシステムの基底に置いてそれを中心にシステム構築を行うのが、より合理的であると考えられる。

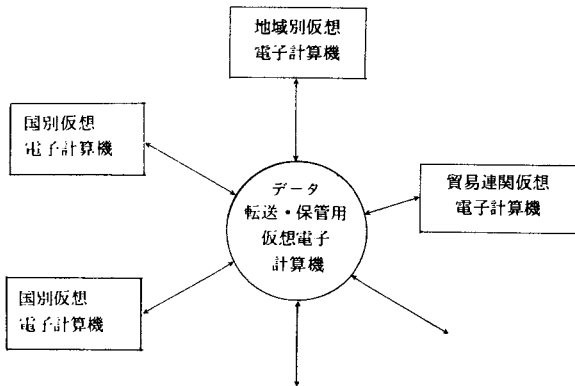
パソコンのソフトが充実したことによって、使い方が比較的簡便でしかもコンパクトで必要かつ十分な機能を備えた、色々なDBMSが利用可能になった。dBASE II， dBASE III， 特に， R:BASE 4000， R:BASE 5000等は、大型機用のものと比べても遜色のないリレーショナル・データベースである。これらはその内臓機能を他のコンパイラ言語、たとえば



第7-3図 ファイルを仮想電子計算機と見た場合の世界計量経済分析システム



第7-4図 データ転送・保管用仮想電子計算機を中心にした場合



FORTRANで書かれたプログラムから利用できるように、プログラム・インターフェースが提供されているので、容易に新しく開発するシステムに組み込むことができる。

PC STEPSの最近の版は、そのデータ・マネージメント・サブシステムを、全面的にR:BASE4000に依存して設計されている。今後の大規模システムは、原則として本格的なDBMSをモジュールの一つとして、組みこんで設計されるべきであると考えられる。この場合、データ管理は、本来のDBMSの諸機能を使ってシステムとは独立に行なうことが出来るので、自由に行き届いたデータ管理体制を作ることが可能である。

計量経済モデル分析の大規模化に伴って、データ管理の不備からくると思われる難点が増大しているようである。観測値データのみでなく、それらを加工して生成したデータも加工手順を含めて統一的にDBMSによって管理することが望ましいし、推定結果やモデルそのものも、やはり、システム独自のファイルではなく、データ・ベースで管理すべきである。FORTRAN等で生成された一般のファイルをデータ・ベース・ファイルに転換する機能は、ほとんどのDBMSに備えられている。

### 7.3 分散型仮想電子計算機

前節までは、これら複合体仮想電子計算機を、単一の汎用電子計算機の上に実現することを暗黙の前提として、その方法について考えて来たが、もしも各々の仮想電子計算機間のデータ転送の手段が何らかの方法で保証されていれば、同一の汎用電子計算機上にある必要はない。これが一般的にいわれる分散型処理システムである。また各々の仮想電子計算機に必要なデータは各々の仮想電子計算機上で管理し処理するシステムが分散型データ・ベース・マネージメント・システムである。

マイクロ・コンピュータの急激な進歩によって、パーソナル・コンピュータ

は今や10年前の大型機に充分匹敵する性能を持つようになった。既に1 MIPS以上の速度とMega単位（100万単位）の記憶容量をもつパソコンが市場にでている。一方、数十・数百のマイコンを結合して平行処理を行なう、安価な超高速電子計算機も近く出現する。大規模な研究計画を、たった一台の大型機を不自由を感じながら他業務の利用者と競合しながら使って、長い期間をかけて実施するなどというのは、およそ非効率という外はない。

大型機の計算料に支払われる金額で、十分な設備をして、はるかに効率的に研究を実施できる状況になったことを理解すべきであり、これに対応しうるソフトウェアの開発を行なうべきである。

#### 7.4 分散型計量経済モデル分析システム

分散型処理システムのイメージを具体的に把握できるように、われわれの構想している分散型世界計量経済モデル分析システムの設計について説明する。このシステムは、京都大学、和歌山大学および神戸大学の共同研究プロジェクトとして、必要設備の設置を急いでおり、近く実用実験を開始する。

各大学および研究参加者自宅には、NECのPC-98XAまたはPC-9801が設置され、互いに交換電話回線（DDX-Pデジタルパケット交換網サービス等）で結ばれてネット・ワークを形成する。大学内でのネット・ワークは、LANになるが、これは大学によって様式が違うのでそれぞれについて別個のソフトを作成することになる。

ネット・ワークに結合されている各電子計算機は、主記憶500KB以上であり、ファイルとして、ディスケット 1 MB \* 2、ハードディスク 10-20 MB、ラムディスク等をもつ。ハードディスク・ラムディスクの容量は後で説明する仮想電子計算機機能のレベルによって異なる。

機種を統一したのは、ソフトウェア開発の負担を軽減するためで、理論的には異機種間ネット・ワークであってもかまわない。実験対象となるデータ転送

は、アスキー・データに限っているので、異機種間ネット・ワークでも重大な支障は生じない。実際にネット・ワークを使って長時間の計算を連続して実行しなければならない段階では、転送しなければならないデータは、内生変数の計算値だけなので、アスキー・データの転送だけで十分であると考ええる。

各電子計算機には、MS. DOS Version 3.1をオペレーティング・システムに、ファイル転送通信ソフト、DBMS、および、PC STEPSが搭載される。DBMSとしては、R:BASE 4000またはR:BASE 5000を使う予定である。通信ソフトは市販のパソコン・ネット・ワーク用ソフトから選んで採用する。

これらは全体として一つのシステムを形成するが、個々の電子計算機について言えば、互いに独立な機能を果たす仮想電子計算機である。もっとも、一台の電子計算機が複数の仮想電子計算機として機能するものと、当然想定されている。

仮想電子計算機は、そのシステムの稼働時に果たす役割によって、

- (1) ワールド・モデル仮想電子計算機
- (2) 地域モデル仮想電子計算機
- (3) 国別モデル仮想電子計算機

の3群に組分けされる。

#### 7.4 1 国別モデル仮想電子計算機群

すべての電子計算機は、国別モデル仮想電子計算機として機能する。国別という場合、モデルがリンクされていない状態では ワールドおよび地域モデルを含めて一つの国と考え、リンクが開始されると、これらは除外するものとする。各国別にモデルの規模が違うので一定しないが、一台が各数カ国を分担する。

国別にモデル名（国名：8字以内）と識別記号（簡略国名：3字以内）とが規定されており、モデル名がファイル識別子となる。変数名には、もしそれが

他国（外部）モデルに属する変数である場合には、前に識別記号を付ける。そのモデルに属する変数であれば識別記号を付けないが、その内容を外部モデルに転送する時は、自動的に変数名に識別記号が付加される。

国別モデルはそれぞれ独立に作成されファイナル・テストを行って、データ、その加工過程、行動方程式等の推定結果、モデル等、すべてDBMSで管理される。

国別モデルの段階においては、各仮想電子計算機は、通常の単一モデルの分析システムである。単一モデルとしての国別モデルの分析を複数の電子計算機で実施することも容易に可能であり、すでに何度か実行されているが、説明が繁雑になるので省略する。

#### 7.4.2 ワールド 地域モデル仮想電子計算機

モデルのリンクが開始されると、一部の電子計算機には、ワールド・モデルまたは地域モデル仮想電子計算機としての機能が割り当てられる。ワールド・モデルおよび地域モデルには演算速度およびファイル容量の関係から、PC-98XAを用いる予定である。もちろん、リンクの処理がおこなわれている過程で処理実施時間に差異があるから、同じ電子計算機を国別モデルと併用してもさしつかえない。

これらの上位モデルの仮想電子計算機はDBMSを持たない。必要なデータはすべて各国別に別の仮想電子計算機で分散管理される。これらの上位モデルの機能は、下位モデルからシミュレーション用データ・ファイルの転送を受けてそれを編集し、シミュレーションを実行して結果を下位モデルに配送することに限られる。

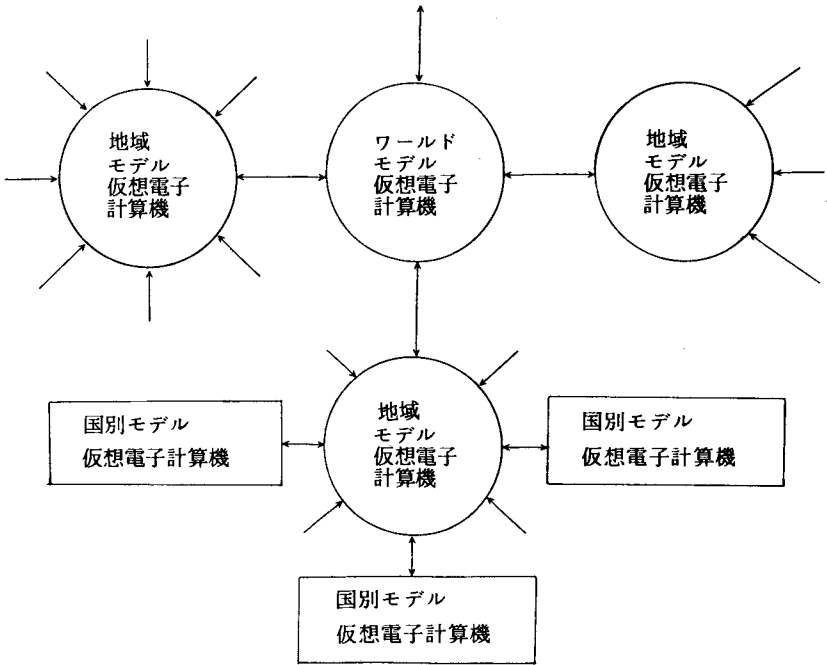
### 7.5 分散型計量経済モデル分析システムの機能

説明を簡単にするため、各国別モデル、地域モデルおよびワールド・モデルが、それぞれ別個の電子計算機で処理されるものとして、分散型計量経済モデ

ル分析システムの機能を述べる。

このシステムは第7-5図に示す構成となる。

第7-5図 分散型計量経済分析システム



リンク・モデルのシミュレーションを実行する場合、ワールド・モデルを担当する電子計算機が、全システムのコントローラとなり、各地域モデルはそれに属する国別モデルのグループについて、サブ・コントローラとなる。東南アジア研究センターのシステムで説明した理由によって、さしづめは、国別モデ

ル相互間、地域モデル相互間のデータ転送は行わず、上位と下位のモデル間のみ限定する。

最初に、最上位モデルであるワールド・モデルからの指令によって、まず、各国別モデルは、自己のシミュレーション用データ・ファイルから地域モデルに対して、自モデルに含まれ、かつ、上位モデル（地域およびワールド）でも用いられる変数名の表、およびそれらの変数の値を、予定されているシミュレーション期間について、データ・ベースから一括転送する。各国別モデルから、その属する地域モデルに、データが送られ、地域モデルは、全ての所属国モデルからのデータ転送が終った段階で、これをCROSS SECTION TYPEに編集して、シミュレーション用データ・ファイルを作成し、下位の国別モデルおよび上位のワールド・モデルに対する、データ・ファイル転送の管理表をつくる。それが終わると、地域モデルはワールド・モデルに、同様のデータ・ファイル転送を行い、全地域モデルからの転送終了を待って、ワールド・モデルの作表作業がなされる。

これらの初期作業が終了すると、リンク・モデルのシミュレーションが開始される。ワールド・モデルから指示された期について、シミュレーションが行われるが、全部の国別シミュレーションが収束するのを待って、地域モデルのシミュレーションが行われ、全部の地域モデルのシミュレーションの収束を待ってワールド・モデルのシミュレーションが行われる。ワールド・モデルが収束すれば、次の期のシミュレーションに移る。

この過程を詳述すると次の様になる。

〔1〕各国別モデルが、それぞれ独立に、平行してシミュレートされ、それぞれ収束すれば結果を地域モデルに転送する。この場合、国別のすべての収束結果が前回地域モデルからおくられてきた結果に対して収束限界値内であれば、シミュレーションは行わず、データをそのまま地域モデルへ送り返す。

〔2〕一つの地域モデルに属する全部の国別モデルが収束してデータが揃えば、

その結果を前回の結果と比較し、収束制御限界値外であれば、地域モデルのシミュレーションが開始され、収束結果は各国別モデルへ転送されて、〔1〕に戻る。

すべての結果が収束限界値内であれば、地域モデルの前回の収束値を、ワールド・モデルに転送して〔3〕に移る。

〔3〕全部の地域モデルが収束すれば、ワールド・モデルはその結果を前回の結果と比較し、収束限界値外であれば、ワールド・モデルのシミュレーションが開始される。収束結果は各地域モデルに転送されて、制御は地域モデルに戻り、〔4〕が実行される。

収束限界値内であれば、モデル全体のシミュレーションは収束したと判定されて、次の期に移り、〔1〕へ戻る。指定された期間が终れば、シミュレーションは完了する。

〔4〕地域モデルは送られてきたワールド・モデルの収束結果を判定して、それぞれ必要な結果を各国別モデルに転送して〔1〕に戻る。もしワールドから送られて来たすべての結果が前回と比較して収束限界値内であれば、転送は行われず、データはそのままワールドへ送り返される。

このシステムでは、国別モデルおよび地域モデルは、それぞれ同時に平行してシミュレーションが行われ、かつ、個々のモデルの一期分のシミュレーションに要する時間はせいぜい数秒であるから、大きいモデルを高速の電子計算機に割り当てる様にすれば、効率よくシミュレーションを実施できると期待している。

単一モデルのシミュレーション中に解が発散する場合は、各内生変数の値をシミュレーション・プログラムがプリントするが、国別モデルや地域モデルが上位モデルの計算終了待ちでアイドル状態にある時に、直前までの結果をプリントするようにしておけば、デバッグしやすいシステムになると考えている。





## 第八章 おわりに

仮想電子計算機という概念を導入することにより計量経済分析システムを電子計算機言語として体系的に分類することを試みた。小規模計量経済分析システムとして単一国モデル分析システム，そして大規模計量経済分析システムとして世界経済モデル分析システムについて，東南アジア研究センターで使用しているシステムを中心に一般的・抽象的な説明と，電子計算機言語としての位置づけについて検討した。

更に近年盛んに議論される様になったデータベース・マネージメント・システムや分散型処理システムについても仮想電子計算機という概念で統一的に解説した。その例としてパーソナル・コンピューターと通信による世界計量経済モデル分析システムの概要について述べた。

計量経済分析だけでなく，多くのシステムが現存するが，これらは全て仮想電子計算機として捉えることによって，その構造を明確にすることができる。つまり小さなシステムでは，単一の仮想電子計算機として，また巨大システムでは色々な機能を持った小さな仮想電子計算機の集合体として，そして更にそれを繋ぐ仮想電子計算機の複合体仮想電子計算機として捉えることが可能であることを示した。

以上の説明で古くからあるシステムから，近年の分散型処理システムや分散型データベースシステムの構成法について統一的な概念を計量経済分析システムを例として示した。この説明を通じて，他の分野のシステムについても同様な概念が展開されることを期待する。

## 参 考 文 献

- [1] 天野明弘 「EPA世界経済モデルの構造」、『経済分析』, 第87号, 1982年10月.
- [2] ———, 栗原英治, Lee Samuelson 「世界経済モデルにおける貿易連関サブ・モデルについて」、『経済分析』, 第80号, 1980年3月.
- [3] Ball, R.J. (ed.), *The international linkage of national economic models*, North-Holland, Amsterdam, 1973.
- [4] Hickman, B.G. (ed.), *Global international economic models*, Springer-Verlag, Amsterdam, 1983.
- [5] Ichimra, S. and Ezaki, M. (ed.), *Econometric Models of Asia Link*, Springer-Verlag, Tokyo, 1985.
- [6] Klein, L.R., "Economic Fluctuations in The United States 1921-1941", *Cowles Commission for Research in Economics Monograph*, No. 11, The University of Chicago, USA, 1950.
- [7] ———, "Five-Year Experience of Linking National Econometric Modes and of Forecasting International Trade", Glejser, H. (ed.), *Quantitative Studies of International Economic Relations*, North-Holland, Amsterdam, 1976.
- [8] 森 敬 「計量経済学とコンピュータⅡ 計量経済モデル分析システムへの要望および機能に関するサーベイと評価」、『情報処理』, Vol.15 No. 2, 1974年 p. 122.
- [9] ———, 「計量経済」, 小野勝章・三浦大亮・森 敬・矢島敬二編 『アプリケーション・プログラム』, bit Vol. 9 No. 9 1977年.
- [10] ———, 『「EMS 計量経済モデルビルディングサポートシステム」利用法』, 通産省大臣官房情報管理課政策情報システム室, 1980年.
- [11] 佐波隆光 『数量経済分析の基礎』, 筑摩書房, 1974年
- [12] Schleicher, S., *STS System: User's Manual*, University of Graz 1980.
- [13] Shishido, S., "Long-term forecasts and policy implications : simulations with a world econometric model (T-FAIS IV)", in Hickman 4.
- [14] Tanenbaum, A.S., *Structured Computer Organization* (Second Edi-

- tion), Prentice-Hall Inc., USA, 1984.
- [15] The University of Tsukuba and IBM Japan, Ltd., "World Econometric Modeling System: (WEMS) —— APL Tool to Build and Solve World Model ——", *Partnership Program World Econometric Model Report*, No. 2, 1981.
- [16] Yasuda, S. "A Program System to Solve Linked Econometric Models", in Ichimura, S. and Ezaki, M.(ed.), *Econometric Models of Asia Link*, Springer-Verlag, Tokyo, 1985. pp.247-254 (Chapter 14)
- [17] 安田 聖 「世界経済モデル分析システム」, 『経済経営研究年報』, 第35号(Ⅱ), 1985.
- [18] Waelgroeck, J., "The Methodology of Linkage", in Ball 3.



付 録

TSP (Time Series Processor) による例

クラインの第一モデル (ファイナルテスト)

TSP VERSION 4.0B  
 COPYRIGHT (C) 1983 TSP INTERNATIONAL  
 ALL RIGHTS RESERVED

IN CASE OF QUESTIONS OR PROBLEMS, SEE YOUR LOCAL TSP CONSULTANT OR SEND A DESCRIPTION OF THE PROBLEM  
 AND THE ASSOCIATED TSP OUTPUT TO:

TSP INTERNATIONAL  
 928 HEARS COURT  
 STANFORD, CA 94305  
 USA

PROGRAM

LINE \*\*\*\*\*

```

1 NAME KLEIN ;
2 LOAD ;
3 SMPL 1 22 ;
4 GENR W=WP+WG ;
5 GENR Y=CN+I+G-T;
6 GENR E=Y+T-WG;
7
8 SMPL 2 22 ;
9
10 PARAM A1,A2,A3,A4;
11 FRML CNO,CN=A1+A2*P+A3*P(-1)+A4*W;
12 LSQ CNO;
13
14 PARAM A5,A6,A7,A8;
15 FRML IO,I=A5+A6*P+A7*P(-1)+A8*K(-1);
16 LSQ IO;
17
18 PARAM A9,A10,A11,A12;
19 FRML WPO,WP=A9+A10*E+A11*E(-1)+A12*TIME;
20 LSQ WPO;
21
22 IDENT YO,Y=CN+I+G-T;
23 IDENT PO,P=Y-W;
24 IDENT WO,W=WP-WG;
25 IDENT EO,E=Y+T-WG;
26 IDENT KO,K=K(-1)*I;
27
28 LIST ENDDGL CN I WP Y P W E K;
29 LIST EQU L CNO IO WPO YO PO WO EO KO;
30 MODEL EQU L,ENDDGL,KLEIN;
31
32 SOLVE (DYNAM) KLEIN;
33
34 STOP ;
35 END ;
    
```



EXECUTION

\*\*\*\*\*

1 1 SMPL 1 22 :

```

CURRENT SAMPLE : 1          TO 22
LOAD CN :
39.8 41.9 45.0 49.2 50.6 52.6 55.1 56.2 57.3 57.8 55.0 50.9 45.6 46.5
48.7 51.3 57.7 58.7 57.5 61.6 65.0 69.7
LOAD I :
2.7 -0.2 1.9 5.2 3.0 5.1 5.6 4.2 3.0 5.1 1.0 -3.4 -6.2 -5.1
-3.0 -1.3 2.1 2.0 -1.9 1.3 3.3 4.9
LOAD WP :
28.8 25.5 29.3 34.1 33.9 35.4 37.4 37.9 39.2 41.3 37.9 34.5 29.0 28.5
30.6 33.2 36.8 41.0 38.2 41.6 45.0 53.3
LOAD P :
15.7 12.4 19.7 18.4 19.4 20.1 19.6 19.8 21.1 21.7 15.6 11.4 7.0 11.2
15.3 14.0 17.6 17.3 15.3 19.0 21.1 23.5
LOAD K :
182.8 182.6 184.5 189.7 192.7 197.8 203.4 207.6 210.6 215.7 216.7
213.3 207.1 202.0 199.0 197.7 199.8 201.8 199.9 201.2 204.5 209.4
LOAD TIME :
-11. -10. -9. -8. -7. -6. -5. -4. -3. -2. -1. 0. 1. 2. 3. 4. 5. 6. 7.
8. 9. 10.
LOAD WB :
2.2 2.7 2.9 2.9 3.1 3.2 3.3 3.6 3.7 4.0 4.2 4.8 5.3 5.6 6.0 6.1 7.4
6.4 6.7 7.8 8.0 8.5
LOAD X :
44.9 45.6 50.1 57.2 57.1 61.0 64.0 64.4 64.5 67.0 61.2 53.4 44.3 45.1
49.7 54.4 62.7 65.0 60.9 69.5 75.7 88.4
LOAD G :
4.6 6.6 6.1 5.7 6.6 6.5 6.6 7.6 7.9 8.1 9.4 10.7 10.2 9.3 10.0 10.5
10.3 11.0 13.0 14.4 15.4 22.3
LOAD T :
3.4 7.7 3.9 4.7 3.8 5.5 7.0 6.7 4.2 4.0 7.7 7.5 8.3 5.4 6.8 7.2 8.3
2.7 7.4 8.9 9.6 11.6
END :
    
```

CURRENT SAMPLE : 1 TO 22

CURRENT SAMPLE : 2 TO 22

NONLINEAR LEAST SQUARES  
\*\*\*\*\*

MAXIMUM NUMBER OF ITERATIONS = 20  
 CONVERGENCE CRITERION = 0.01000  
 STEPSIZE METHOD = BARD  
 MAXIMUM NUMBER OF STEPSIZE ITERATIONS (MAXSQZ) = 10

EQUATIONS: CNO

NOTE => THE MODEL IS LINEAR IN THE PARAMETERS.  
 WORKING SPACE USED BY LSQ = 732

STARTING CONDITIONS FOR ESTIMATION:

	A1	A2	A3	A4
VALUE	0.00000	0.00000	0.00000	0.00000

ITERATION NUMBER 1

\*\*\*\*\*  
 GRADIENT: 62148.74 OLD STEPSIZE: 1.000000 OLD F: 0.1787945D+02 NEW STEPSIZE: 0.2500000  
 F = 62167. FNEW = 17.879 ISQZ = 0 STEPSIZE = 1.0000 CRITERION = 62149.

	A1	A2	A3	A4
ESTIMATE	0.00000	0.00000	0.00000	0.00000
CHANGES	16.23660	0.19293	0.08988	0.79622

CONVERGENCE ACHIEVED BY GAUSS AFTER 1 ITERATIONS.

4 FUNCTION EVALUATIONS

EQUATION CNO  
\*\*\*\*\*

DEPENDENT VARIABLE: CN

SUM OF SQUARED RESIDUALS = 17.8794  
 STANOARD ERROR OF THE REGRESSION = 1.02354  
 MEAN OF DEPENDENT VARIABLE = 53.9952  
 STANDARD DEVIATION = 6.86087  
 R-SQUARED = 0.981008  
 ADJUSTED R-SQUARED = 0.977657  
 DURBIN-WATSON STATISTIC (ADJ. FOR 0 GAPS) = 1.3675  
 NUMBER OF OBSERVATIONS = 21  
 F-STATISTIC ( 3, 17) = 292.708  
 LOG OF LIKELIHOOD FUNCTION = -28.1086

PARAMETER	ESTIMATE	STANDARD ERROR	T-STATISTIC
A1	16.23660	1.302698	12.46383
A2	0.1929347	0.9121022D-01	2.115275
A3	0.8988457D-01	0.9064800D-01	0.9915781
A4	0.7962188	0.3994392D-01	19.93342

NONLINEAR LEAST SQUARES  
\*\*\*\*\*

MAXIMUM NUMBER OF ITERATIONS = 20  
 CONVERGENCE CRITERION = 0.01000  
 STEP SIZE METHOD = BARD  
 MAXIMUM NUMBER OF STEP SIZE ITERATIONS (MAXSQZ) = 10  
 EQUATIONS: IO

NOTE => THE MODEL IS LINEAR IN THE PARAMETERS.  
 WORKING SPACE USED BY LSQ = 732

STARTING CONDITIONS FOR ESTIMATION:

	A5	A6	A7	A8
VALUE	0.00000	0.00000	0.00000	0.00000

ITERATION NUMBER 1  
 \*\*\*\*\*  
 GRADIENT: 268.6972 OLD STEP SIZE: 1.000000 OLD F: 0.17322710+02 NEW STEP SIZE: 0.2500000  
 F = 266.02 FNEW = 17.323 ISQZ = 0 STEP SIZE = 1.0000 CRITERION = 268.70

	A5	A6	A7	A8
ESTIMATE	0.00000	0.00000	0.00000	0.00000
CHANGES	10.12580	0.47964	0.33304	-0.11179

CONVERGENCE ACHIEVED BY GAUSS AFTER 1 ITERATIONS.  
 4 FUNCTION EVALUATIONS

EQUATION 10  
\*\*\*\*\*

DEPENDENT VARIABLE: I

SUM OF SQUARED RESIDUALS = 17.3227  
 STANDARD ERROR OF THE REGRESSION = 1.00945  
 MEAN OF DEPENDENT VARIABLE = 1.26667  
 STANDARD DEVIATION = 3.55195  
 R-SQUARED = 0.931348  
 ADJUSTED R-SQUARED = 0.919233  
 DURBIN-WATSON STATISTIC (ADJ FOR 0 GAPS) = 1.8102  
 NUMBER OF OBSERVATIONS = 21  
 F-STATISTIC (3, 17) = 16.8753  
 LOG OF LIKELIHOOD FUNCTION = -27.7764

PARAMETER	ESTIMATE	STANDARD ERROR	T-STATISTIC
A5	10.12580	5.465547	1.852660
A6	0.4796355	0.9711462D-01	4.938860
A7	0.3330393	0.1008593	3.302018
A8	-0.1117948	0.2672757D-01	-4.182752

NONLINEAR LEAST SQUARES  
\*\*\*\*\*

MAXIMUM NUMBER OF ITERATIONS = 20  
 CONVERGENCE CRITERION = 0.01000  
 STEPSIZE METHOD = BARD  
 MAXIMUM NUMBER OF STEPSIZE ITERATIONS (MAXSQZ) = 10

EQUATIONS: WPO

NOTE => THE MODEL IS LINEAR IN THE PARAMETERS.  
 WORKING SPACE USED BY LSQ = 732

STARTING CONDITIONS FOR ESTIMATION:

	A9	A10	A11	A12
VALUE	0.00000	0.00000	0.00000	0.00000

ITERATION NUMBER 1

\*\*\*\*\*  
 GRADIENT: 28550.85 OLD STEPSIZE: 1.000000 OLD F: 0.10004720+02 NEW STEPSIZE: 0.2500000  
 F = 28561. FNEW = 10.005 ISQZ = 0 STEPSIZE = 1.0000 CRITERION = 28551.

	A9	A10	A11	A12
ESTIMATE	0.00000	0.00000	0.00000	0.00000
CHANGES	1.49705	0.43948	0.14609	0.13025

CONVERGENCE ACHIEVED BY GAUSS AFTER 1 ITERATIONS.

4 FUNCTION EVALUATIONS

EQUATION WPO  
\*\*\*\*\*

DEPENDENT VARIABLE: WP

SUM OF SQUARED RESIDUALS = 10.0047  
STANDARD ERROR OF THE REGRESSION = 0.767144  
MEAN OF DEPENDENT VARIABLE = 36.3619  
STANDARD DEVIATION = 4.30440  
R-SQUARED = 0.987414  
ADJUSTED R-SQUARED = 0.985193  
DURBIN-WATSON STATISTIC (ADJ. FOR 0 GAPS) = 1.9584  
NUMBER OF OBSERVATIONS = 21  
F-STATISTIC (3, 17) = 444.569  
LOG OF LIKELIHOOD FUNCTION = -22.0123

PARAMETER	ESTIMATE	STANDARD ERROR	T-STATISTIC
A9	1.497050	1.270030	1.178752
A10	0.4394767	0.3240756D-01	13.56093
A11	0.1460902	0.3742310D-01	3.903745
A12	0.1302455	0.3191026D-01	4.081619

ORDERING INFORMATION

THE EQUATIONS ARE TO BE SOLVED IN THE FOLLOWING ORDER:

	EQUATION	DEPENDENT VARIABLE
1	1 CNO	1 CN
2	2 IO	2 I
3	3 WPO	3 WP
4	4 YO	4 Y
5	5 PO	5 P
6	6 WD	6 W
7	7 EO	7 E
8	8 KO	8 K

BLOCK #	# RECURSIVE EQUATIONS	# SIMULTANEOUS EQUATIONS	LINEAR BLOCK?
1	0	7	F
2	1	0	F

THE ORDERED MODEL WAS STORED UNDER THE NAME KLEIN . IT IS 46 WORDS LONG.

## SIMULATION OF THE MODEL KLEIN

\*\*\*\*\*

## OPTIONS FOR THIS ROUTINE

\*\*\*\*\*

CONV1	=	0.01000	CONV2	=	0.00100	OEBUG	=	FALSE
DYNAM	=	TRUE	ITERMX	=	50	KILL	=	TRUE
MAXIT	=	50	MAXPRT	=	5	METHOD	=	GAUSS
PRINT	=	FALSE	PRNDAT	=	FALSE	PRNRES	=	FALSE
PRNSIM	=	TRUE	RESIDU	=	FALSE	SOLNAM	=	
TAG	=		TOL	=	0.01000			

NUMBER OF EQUATIONS IN THE MODEL = 8

NUMBER OF BLOCKS IN THE MODEL = 2

BLOCK # NUMBER OF EQUATIONS

1	7
2	1

PERIOD: 2	BLOCK	1	CONVERGED.
PERIOD: 3	BLOCK	1	CONVERGED.
PERIOD: 4	BLOCK	1	CONVERGED.
PERIOD: 5	BLOCK	1	CONVERGED.
PERIOD: 6	BLOCK	1	CONVERGED.
PERIOD: 7	BLOCK	1	CONVERGED.
PERIOD: 8	BLOCK	1	CONVERGED.
PERIOD: 9	BLOCK	1	CONVERGED.
PERIOD: 10	BLOCK	1	CONVERGED.
PERIOD: 11	BLOCK	1	CONVERGED.
PERIOD: 12	BLOCK	1	CONVERGED.
PERIOD: 13	BLOCK	1	CONVERGED.
PERIOD: 14	BLOCK	1	CONVERGED.
PERIOD: 15	BLOCK	1	CONVERGED.
PERIOD: 16	BLOCK	1	CONVERGED.



PERIOD: 17 BLOCK 1 CONVERGED.  
 PERIOD: 18 BLOCK 1 CONVERGED.  
 PERIOD: 19 BLOCK 1 CONVERGED.  
 PERIOD: 20 BLOCK 1 CONVERGED.  
 PERIOD: 21 BLOCK 1 CONVERGED.  
 PERIOD: 22 BLOCK 1 CONVERGED.

SIMULATION RESULTS

	CN	I	WP	Y	P	W	E	K
2	43.91493	-0.21441	27.66883	42.59022	12.22535	30.36486	47.59022	182.58558
3	48.27104	3.09427	31.25344	53.55611	19.40698	34.14913	54.55611	185.68185
4	52.63581	6.07368	35.45310	59.70019	21.35047	38.34952	61.50019	191.75553
5	56.76582	7.64618	39.41088	67.20115	24.69445	42.50670	67.90115	199.40172
6	56.52904	6.02054	39.57955	63.56079	20.77692	42.78386	65.86079	205.42226
7	50.36357	0.17475	34.13276	50.14887	12.71205	37.43682	53.84887	205.59701
8	44.77632	-4.05980	28.49889	41.62598	9.52345	32.10253	44.72598	201.53720
9	45.83963	-1.99367	28.75089	47.53565	15.08873	32.44692	48.03565	199.54354
10	51.89690	2.76494	34.07459	58.75282	20.68170	38.07112	58.75282	202.30848
11	54.60607	2.75086	37.43803	59.04714	17.41289	41.63426	62.24714	205.05934
12	54.77395	0.83846	37.67162	58.82100	16.34608	42.47493	61.52101	205.89780
13	52.08054	-1.64578	34.93727	52.34392	12.10312	40.24080	55.34392	204.25202
14	50.82129	-1.82051	33.00453	52.90311	14.29752	38.60559	52.70311	202.43151
15	52.19999	-0.67358	33.98563	54.71702	14.75501	39.98202	55.51702	201.75793
16	53.47158	-0.37436	35.39345	56.38866	14.89850	41.49015	57.48866	201.38357
17	52.83618	-2.02768	34.15426	52.81718	11.25959	41.55760	53.71718	199.35589
18	52.91351	-1.50241	34.60586	55.70213	14.39972	41.30241	55.70213	197.85348
19	58.93033	2.00283	39.65037	66.52437	19.17739	47.34699	66.22437	199.85632
20	64.13414	4.18699	45.13491	73.80992	20.87933	52.93059	74.90992	204.04330
21	66.68721	4.17790	48.00368	76.65402	20.65461	55.99941	78.25402	208.22121
22	75.38570	7.27036	56.61736	93.34586	28.23243	65.11343	96.44586	215.49157

\*\*\*\*\*

END OF OUTPUT FOR USER KLEIN  
 WORKING SPACE AVAILABLE IN WORDS = 118382

STS (Stochastic Simulation System) による例

クライアントの第一モデル (ファイナルテスト)



S T S - S Y S T E M      KOBE UNIVERSITY HITAC-M240D (V5.0)

STOCHASTIC SIMULATION SYSTEM      86-01-23 17:22:28

OK

--> \*FILE, I KLEIN

OK

--> \*DATA, I C \*1920 \*1941

? 1920

--> 39.8 41.9 45.0 49.2 50.6 52.6 55.1 56.2 57.3 57.8

? 1930

--> 55.0 50.9 45.6 46.5 48.7 51.3 57.7 58.7 57.5 61.6

? 1940

--> 65.0 69.7

          INSERTED DATA ELEMENT KLEIN.C                      FROM 1920            TO 1941

--> \*DATA, I I \*1920 \*1941

? 1920

--> 2.7 -0.2 1.9 5.2 3.0 5.1 5.6 4.2 3.0 5.1 1.0 -3.4 -6.2 -5.1

? 1934

--> -3.0 -1.3 2.1 2.0 -1.9 1.3 3.3 4.9

          INSERTED DATA ELEMENT KLEIN.I                      FROM 1920            TO 1941

--> \*DATA, I WP \*1920 \*1941

? 1920

--> 28.8 25.5 29.3 34.1 33.9 35.4 37.4 37.9 39.2 41.3

? 1930

--> 37.9 34.5 29.0 28.5 30.6 33.2 36.8 41.0 38.2 41.6

? 1940

--&gt; 45.0 53.3

INSERTED	DATA ELEMENT	KLEIN.WP	FROM 1920	TO 1941
----------	--------------	----------	-----------	---------

--&gt; \*DATA, I P \*1920 \*1941

? 1920

--&gt; 12.7 12.4 16.9 18.4 19.4 20.1 19.6 19.8 21.1 21.7

? 1930

--&gt; 15.6 11.4 7.0 11.2 12.3 14.0 17.6 17.3 15.3 19.0

? 1940

--&gt; 21.1 23.5

INSERTED	DATA ELEMENT	KLEIN.P	FROM 1920	TO 1941
----------	--------------	---------	-----------	---------

--&gt; \*DATA, I K \*1920 \*1941

? 1920

--&gt; 182.8 182.6 184.5 189.7 192.7 197.8 203.4 207.6

? 1928

--&gt; 210.6 215.7 216.7 213.3 207.1 202.0 199.0 197.7

? 1936

--&gt; 199.8 201.8 199.9 201.2 204.5 209.4

INSERTED	DATA ELEMENT	KLEIN.K	FROM 1920	TO 1941
----------	--------------	---------	-----------	---------

--&gt; \*DATA, I WG \*1920 \*1941

? 1920

--&gt; 2.2 2.7 2.9 2.9 3.1 3.2 3.3 3.6 3.7 4.0 4.2 4.8 5.3

? 1933

&gt; 5.6 6.0 6.1 7.4 6.7 7.7 7.8 8.0 8.5

INSERTED	DATA ELEMENT	KLEIN.WG	FROM 1920	TO 1941
----------	--------------	----------	-----------	---------

--&gt; \*DATA, I G \*1920 \*1941

? 1920

--&gt; 4.6 6.6 6.1 5.7 6.6 6.5 6.6 7.6 7.9 8.1 9.4 10.7 10.2

? 1933

--&gt; 9.3 10.0 10.5 10.3 11.0 13.0 14.4 15.4 22.3

```

      INSERTED DATA ELEMENT KLEIN.G          FROM 1920    TO 1941

--> *DATA, I T *1920 *1941

? 1920
--> 3.4 7.7 3.9 4.7 3.8 5.5 7.0 6.7 4.2 4.0 7.7 7.5 8.3
? 1933
--> 5.4 6.8 7.2 8.3 6.7 7.4 8.9 9.6 11.6
      INSERTED DATA ELEMENT KLEIN.T          FROM 1920    TO 1941

--> *DATA, I TIME *1920 *1941

? 1920
--> -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10
      INSERTED DATA ELEMENT KLEIN.TIME       FROM 1920    TO 1941

--> *DATA, I X *1920 *1941

? 1920
--> 44.9 45.6 50.1 57.2 57.1 61.0 64.0 64.4 64.5 67.0 61.2 53.4
? 1932
--> 44.3 45.1 49.7 54.4 62.7 65.0 60.9 69.5 75.7 88.4
      INSERTED DATA ELEMENT KLEIN.X          FROM 1920    TO 1941

--> *DATA, I W *1920 *1941

? 1920
--> 31.0 28.2 32.2 37.0 37.0 38.6 40.7 41.5 42.9 45.3 42.1
? 1931
--> 39.3 34.3 34.1 36.6 39.3 44.2 47.7 45.9 49.4 53.0 61.8
      INSERTED DATA ELEMENT KLEIN.W          FROM 1920    TO 1941

--> *DATA, I Y *1920 *1941
--> <C+I+G-T>

      INSERTED DATA ELEMENT KLEIN.Y          FROM 1920    TO 1941

```

128

--> \*DATA, I E \*1920 \*1941  
--> <Y+T-WG>

INSERTED DATA ELEMENT KLEIN.E FROM 1920 TO 1941

--> \*

\*\*\* INVALID SPECIFICATION \*\*\*

--> \*TIME 1921 1941

OK

--> \*EQU, I C

--> C=F(P, P!11, W)

DBANK STS KLEIN

C

VARIABLE DESCRIPTION

C  
P  
W

PAR.	ESTIMATE	ST. DEV.	T	BC %	VARIABLE
R2	0.981	R2C 0.978	MAPE 1.28		C
1	0.19293	0.09121	2.12	11.3	P

2	0.08989	0.09065	0.99	5.0	P111
3	0.79622	0.03994	19.93	83.6	W
4	16.23659	1.30270	12.46		CONST

```
-----
SE      1.02554   DW 1.37   RHO(1) 0.25   1921   -   1941
-----
```

INSERTED EQU ELEMENT KLEIN.C

--> \*EQU, I I

--> I=F(P, P11!, K11!)

DBANK STS KLEIN

I

VARIABLE DESCRIPTION

I

P

K

```
-----
R2 0.931   R2C 0.919   MAPE 24.78           I
```

```
-----
PAR.  ESTIMATE  ST.DEV.  T    BC x  V A R I A B L E
-----
```

I	0.47964	0.09711	4.94	45.2	P
---	---------	---------	------	------	---



130

2	0.33304	0.10086	3.30	30.0	P!!!
3	-0.11179	0.02673	4.18	24.8	K!!!
4	10.12579	5.46555	1.85		CONST

```
-----
SE      1.00945    DW 1.81    RHO(1) 0.08    1921    -    1941
=====
```

INSERTED EQU ELEMENT KLEIN. I

--> \*EQU, I WP

--> WP=F(E, E111, TIME)

DBANK STS KLEIN

WP

VARIABLE DESCRIPTION

WP  
E  
TIME

```
-----
R2 0.987    R2C 0.985    MAPE 1.60            WP
-----
PAR.  ESTIMATE    ST.DEV.    T    BC %    V A R I A B L E
-----
1      0.43948    0.03241    13.56    68.8    E
```

2	0.14609	0.03742	3.90	19.2	E111
3	0.13025	0.03191	4.08	11.9	TIME
4	1.49704	1.27003	1.18		CONST

```
-----
SE      0.76715      DW 1.96      RHO(1)-0.08      1921      1941
-----
```

INSERTED EQU ELEMENT KLEIN.WP

--> \*EQU, I Y

--> Y=C+I+G-T

INSERTED EQU ELEMENT KLEIN.Y

--> \*EQU, I P

--> P=Y-W

INSERTED EQU ELEMENT KLEIN.P

--> \*EQU, I W

--> W=WP+WG

INSERTED EQU ELEMENT KLEIN.W

--> \*EQU, I E

--> E=Y+T-WG

INSERTED EQU ELEMENT KLEIN.E

--> \*EQU, I K

132

--> K=K!!!+I

--> \*

INSERTED EQU ELEMENT KLEIN.K

--> \*FILE, I KLEIN10

OK

--> \*COPY, D KLEIN. , KLEIN10.

DATA ELEMENT KLEIN. C	COPIED TO KLEIN10. C
DATA ELEMENT KLEIN. I	COPIED TO KLEIN10. I
DATA ELEMENT KLEIN. WP	COPIED TO KLEIN10. WP
DATA ELEMENT KLEIN. P	COPIED TO KLEIN10. P
DATA ELEMENT KLEIN. K	COPIED TO KLEIN10. K
DATA ELEMENT KLEIN. WG	COPIED TO KLEIN10. WG
DATA ELEMENT KLEIN. G	COPIED TO KLEIN10. G
DATA ELEMENT KLEIN. T	COPIED TO KLEIN10. T
DATA ELEMENT KLEIN. TIME	COPIED TO KLEIN10. TIME
DATA ELEMENT KLEIN. X	COPIED TO KLEIN10. X
DATA ELEMENT KLEIN. W	COPIED TO KLEIN10. W
DATA ELEMENT KLEIN. Y	COPIED TO KLEIN10. Y
DATA ELEMENT KLEIN. E	COPIED TO KLEIN10. E

--> \*FILE, B KLEIN

OK

--> \*FILE, D KLEIN10

OK

--> \*FILE, S KLEIN

OK

--> \*FILE, E KLEIN

OK

--> \*SYS, IS KLEIN

PLEASE INPUT THE EQUATION ELEMENT NAME

--> C

--> I

--> WP

--> Y

--> P

--> W

--> E

--> K

--> \*

\*\*\* SYSTEM STRUCTURE ANALYSIS START. \*\*\*

DO YOU NEED A LINE PRINTER OUTPUT ? YES/NO

NO

DO YOU NEED A TERMINAL OUTPUT ? YES/NO

YES

S T S - S Y S T E M

    K O B E

S Y S T E M   S T R U C T U R E   A N A L Y S I S

KLEIN

I N P U T   S Y S T E M

1   C           P           W

2   I           P

3	WP	E	
4	Y	C	I
5	P	Y	W
6	W	WP	
7	E	Y	
8	K	I	

1            5

C I W Y P W E K

    P

1	C	*		*	*
2	I		*		*
3	WP		*		*
4	Y	*	*	*	
5	P			*	*
6	W		*		*
7	E			*	*
8	K	*			*

B L O C K R E C U R S I V E   S Y S T E M

2 B L O C K S

BLOCK	1	1	E	7
		2	WP	3
		3	W	6
		4	I	2

		5	Y		4
		6	P		5
		7	C		1
BLOCK	2	8	K		8

1            5

E W W I Y P C K

P

1	E	*		*			
2	WP	*	*				
3	W		*	*			
4	I			*		*	
5	Y			*	*		*
6	P		*		*	*	
7	C		*			*	*
8	K			*			*

INSERTED	SYS ELEMENT	KLEIN.KLEIN
SOLUTION FOR	1921	24 ITERATIONS
SOLUTION FOR	1922	18 ITERATIONS
SOLUTION FOR	1923	16 ITERATIONS
SOLUTION FOR	1924	17 ITERATIONS
SOLUTION FOR	1925	16 ITERATIONS
SOLUTION FOR	1926	30 ITERATIONS
SOLUTION FOR	1927	22 ITERATIONS
SOLUTION FOR	1928	24 ITERATIONS

SOLUTION FOR 1929	20 ITERATIONS
SOLUTION FOR 1930	12 ITERATIONS
SOLUTION FOR 1931	24 ITERATIONS
SOLUTION FOR 1932	23 ITERATIONS
SOLUTION FOR 1933	21 ITERATIONS
SOLUTION FOR 1934	23 ITERATIONS
SOLUTION FOR 1935	24 ITERATIONS
SOLUTION FOR 1936	21 ITERATIONS
SOLUTION FOR 1937	22 ITERATIONS
SOLUTION FOR 1938	19 ITERATIONS
SOLUTION FOR 1939	16 ITERATIONS
SOLUTION FOR 1940	13 ITERATIONS
SOLUTION FOR 1941	17 ITERATIONS

DO YOU REPLACE THE DATA ELEMENT(S) IN KLEIN10 FILE ?  
PLEASE TYPE IN --> YES/NO

--> \*REP, ALL KLEIN

--> \*END

STS-SYSTEM TERMINATED 86-01-23 17:22:49

VARIABLE NAME      DESCRIPTION      UNIT      SOURCE

ENDOGENOUS VARIABLES

E  
W  
P  
Y  
C  
K

EXOGENOUS VARIABLES

T  
G  
I  
W  
G

YEAR	E	WP	W	I	Y	P	C	K	TIME	G
1920	44.9000	28.8000	31.0000	2.70000	43.7000	12.7000	39.8000	182.800	-11.0000	4.60000
1921	47.6136	27.6791	30.3791	-2.212379	42.6143	12.2352	43.9271	182.588	-10.0000	6.60000
1922	54.5607	31.2589	34.1589	3.09698	53.5709	19.4120	48.2795	185.685	-9.00000	6.10000
1923	61.4535	35.4333	38.3333	6.06375	59.6732	21.3400	52.6203	191.748	-8.00000	5.70000
1924	67.8066	39.5025	42.4625	7.62419	67.1335	24.6710	56.7240	199.372	-7.00000	6.60000
1925	69.7334	39.5097	42.7097	5.92810	63.4507	20.7410	56.4621	205.371	-6.00000	6.50000
1926	53.7809	34.0842	37.3842	0.163376	50.0803	12.6961	50.3164	205.534	-5.00000	6.60000
1927	44.7437	28.4847	32.0947	-4.04174	41.6321	9.55337	44.7735	201.484	-4.00000	7.60000
1928	48.1244	28.7925	32.4925	-1.96744	47.6188	15.1263	45.8832	199.519	-3.00000	7.90000
1929	58.8818	34.1442	38.1442	2.80249	58.8737	20.7295	51.9668	202.521	-2.00000	8.10000
1930	62.6174	37.4878	41.6878	2.77326	59.1256	17.4378	54.6568	205.095	-1.00000	9.40000
1931	61.5179	37.6805	42.4805	0.841462	58.8204	16.3399	54.7803	205.936	0.0	10.7000
1932	55.2764	34.9071	40.2071	-1.66479	52.2811	12.0739	52.0484	204.271	1.00000	10.2000
1933	52.6170	32.9568	38.5568	-1.84832	52.8225	14.2657	50.7738	202.423	2.00000	9.30000
1934	55.4798	35.9567	39.9567	-6.90564	54.6823	14.7256	52.1742	201.732	3.00000	10.0000
1935	57.4998	35.3929	41.4929	-3.72444	56.4009	14.9080	53.4739	201.360	4.00000	10.5000
1936	55.7492	34.1700	41.5679	-2.01244	52.8417	11.2718	52.8501	199.348	5.00000	10.3000
1937	55.7653	34.6401	41.3401	-1.48688	55.7644	14.4243	52.9485	197.861	6.00000	11.0000
1938	66.2633	39.6773	47.3773	2.01381	66.5692	19.1619	58.9586	199.874	7.00000	13.0000
1939	74.8985	45.1354	52.9355	4.18328	73.8132	20.8776	64.1379	204.058	8.00000	14.4000
1940	78.2128	47.9839	55.9839	4.16497	76.6281	20.6443	66.6716	208.223	9.00000	15.4000
1941	96.3575	56.5725	65.0725	7.24833	93.2817	28.2092	75.3467	215.471	10.0000	22.3000



YEAR	T	WG
1920	3.40000	2.20000
1921	7.70000	2.70000
1922	3.90000	2.90000
1923	4.70000	2.90000
1924	4.80000	3.10000
1925	5.50000	3.20000
1926	7.00000	3.30000
1927	6.70000	3.60000
1928	4.20000	3.70000
1929	4.00000	4.00000
1930	7.70000	4.20000
1931	7.50000	4.80000
1932	8.30000	5.30000
1933	5.40000	5.60000
1934	6.80000	6.00000
1935	7.20000	6.10000
1936	6.50000	7.40000
1937	6.70000	6.70000
1938	7.40000	7.70000
1939	8.90000	7.80000
1940	9.60000	8.00000
1941	11.6000	8.50000

1.

E = Y + T - WG

2.

WP = 1.4970 + 0.4394 \* E + 0.1460 \* E111 + 0.1302 \* TIME  
 (1.17) (13.56) (3.90) (4.08)

DW/SE/R2 1.9584/0.7671/0.9851 1921 - 1941

3.

W = WP + WG

4.

I = 10.1257 + 0.4796 \* P + 0.3330 \* P111 - 0.1117 \* K111  
 (1.85) (4.93) (3.30) (4.18)

DW/SE/R2 1.8101/1.0094/0.9192 1921 - 1941

5.

Y = C + I + G - T

6.

P = Y - W

7.

C = 16.2365 + 0.1929 \* P + 0.08988 \* P111 + 0.7962 \* W  
 (12.46) (2.17) (0.99) (19.93)

DW/SE/R2 1.3674/1.0255/0.9776 1921 - 1941

8.

K = K111 + I

仮想電子計算機と計算機言語システム  
— 世界計量経済モデル分析システム —

---

昭和61年6月10日 印刷

(非売品)

昭和61年6月20日 発行

著 作 神戸大学助教授  
やす だ さとし  
安 田 聖

発行所 神戸市灘区六甲台町  
神戸大学経済経営研究所

印刷所 神戸市中央区中山手通7丁目5番7号  
有限会社 興 文 社

---